

### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

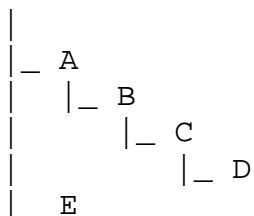
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 1



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

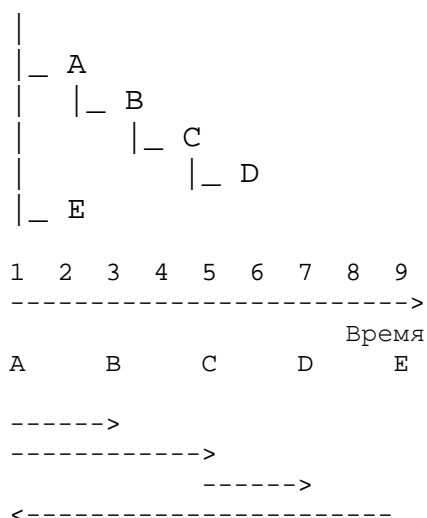
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 1



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

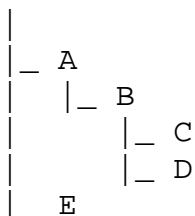
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 2



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

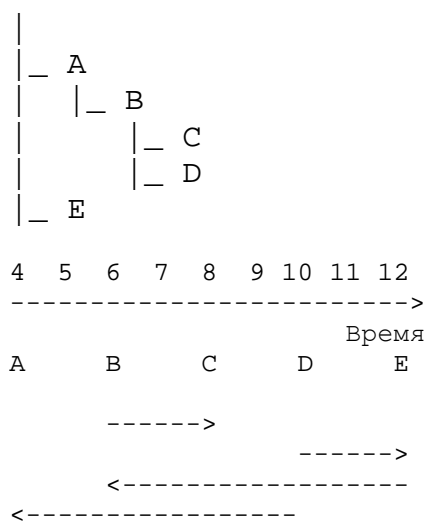
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 2





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

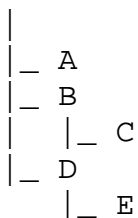
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 3



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWrite String свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

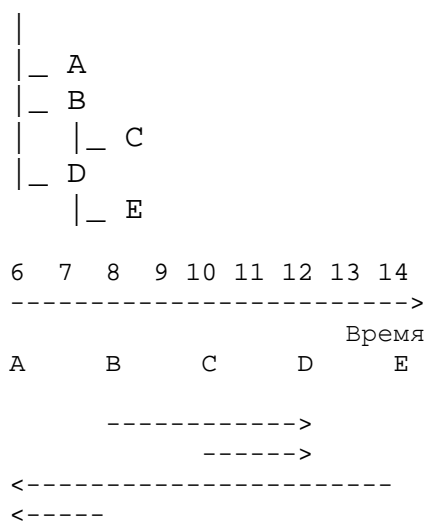
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана Ом и Поток событий некоторой ЭС. Построить ПМ Ом ЭС.  
 Вариант 3



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

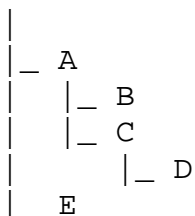
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 4



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

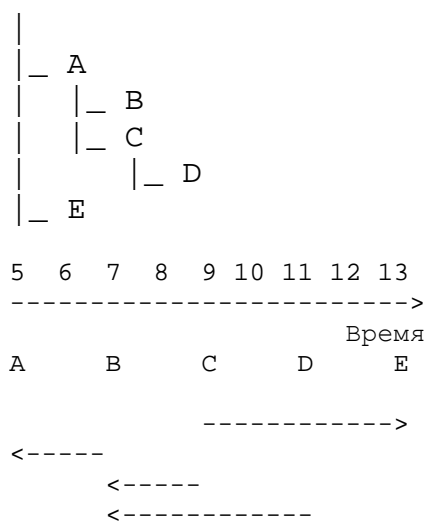
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 4





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

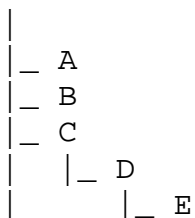
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 5



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

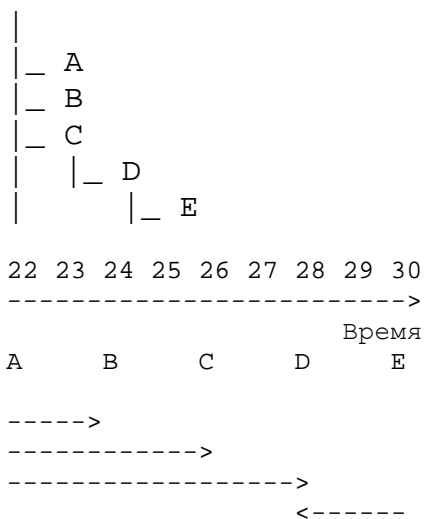
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

**Task\_09\_EventDynamic**

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 5



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 7

```
|
|_ A
|_ B
    |_ C
    |_ D
    |_ E
```

Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента.
- Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

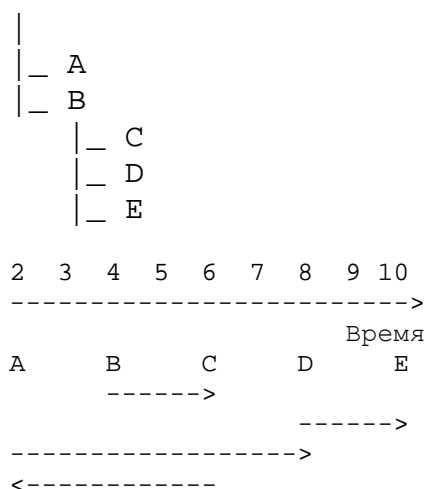
1 Отображение значений свойств выполнить оператором MsgBox.

2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.

Вариант 7





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

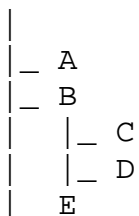
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 9



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

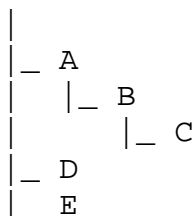
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 11



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

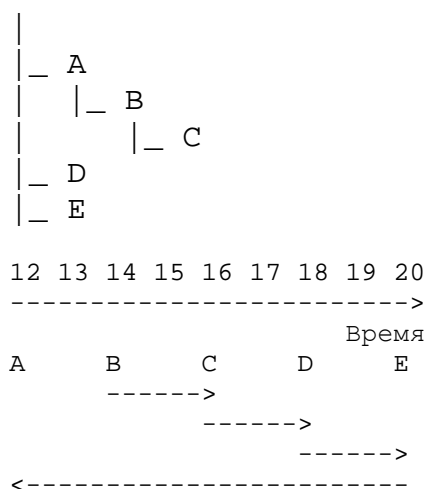
Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.

Вариант 11





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

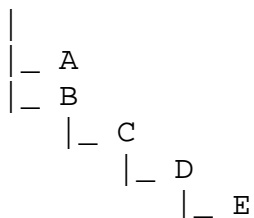
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 12



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

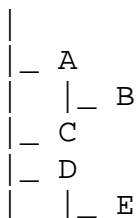
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 14



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 15

```
|
|_ A
|   |_ B
|_ C
|_ D
|_ E
```

Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

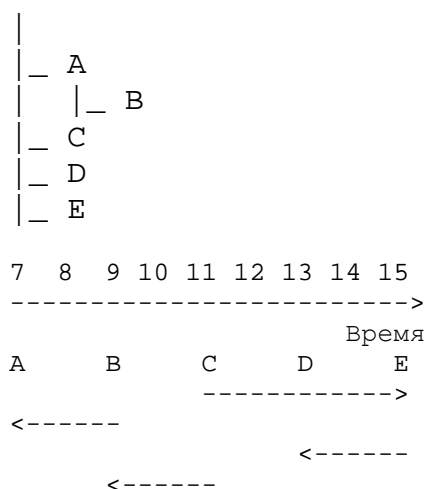
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 15



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

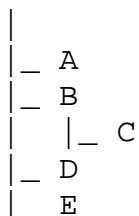
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 16



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

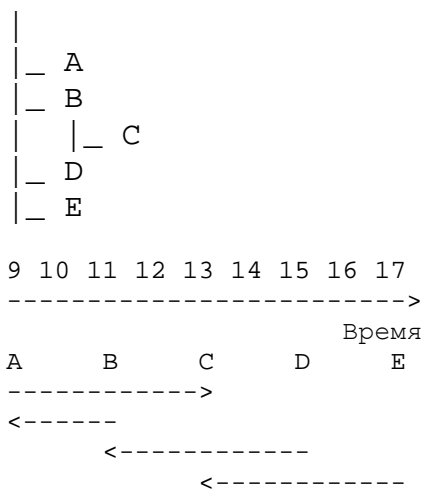
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

**Task\_09\_EventDynamic**

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
 Вариант 16





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

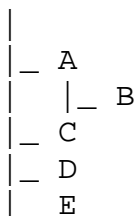
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 17



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

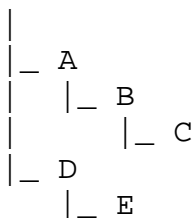
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 18



Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет Protected метод MMM, отображающий "Это защищенный метод базового класса".

а) В Main методе производного класса самого нижнего уровня иерархии вызвать Protected метод MMM.

б) В Main методе класса, находящегося вне иерархии, вызвать Protected метод MMM.

2) Пусть класс верхнего уровня иерархии имеет Protected ReadWrite String типа свойство PPP, значение которого инициализируется при создании экземпляра класса величиной «xxxxxx».

В Main методе класса, находящегося вне иерархии

- a) Отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «wwwww»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента.

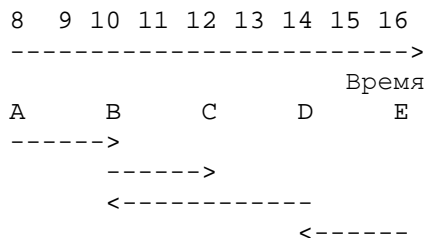
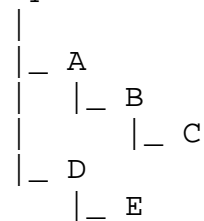
Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

### Task\_09\_EventDynamic

Дана OM и Поток событий некоторой ЭС. Построить ПМ OM ЭС.

Вариант 18

SystemX





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

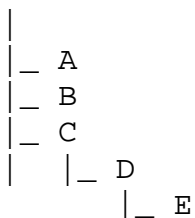
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 19



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

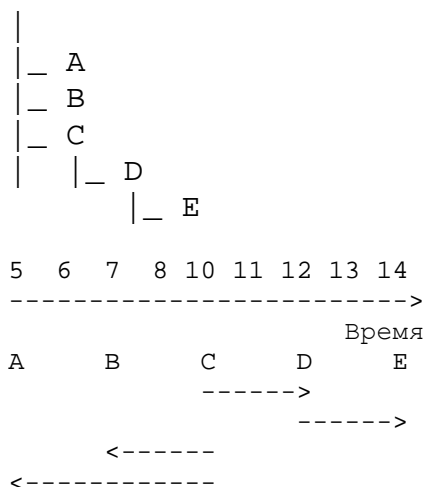
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 19



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 20

```
|
|_ A
|_ B
|_ C
|_ D
|   |_ E
```

Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

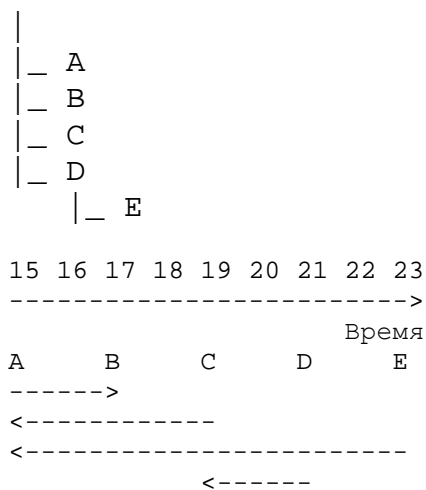
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

**Task\_09\_EventDynamic**

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 20





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

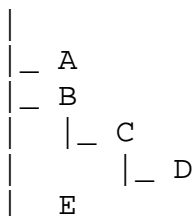
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 22



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

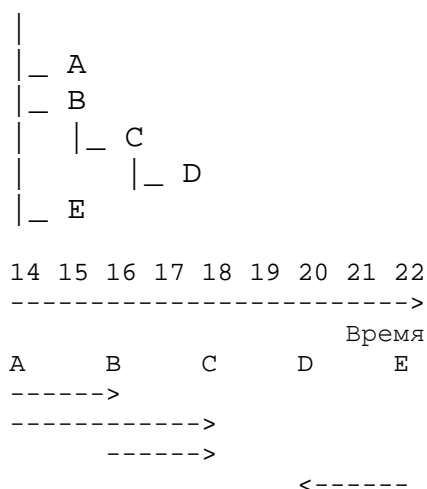
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 22



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

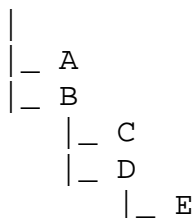
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 24



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

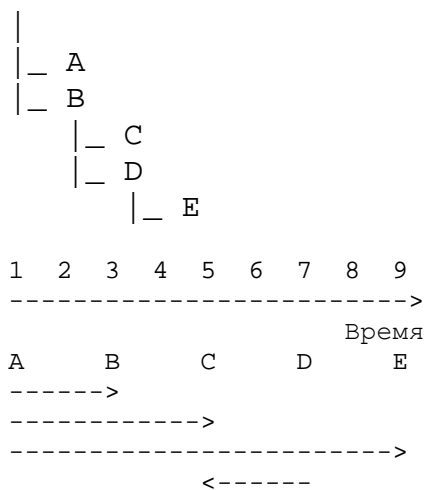
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

**Task\_09\_EventDynamic**

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 24





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 25

```
|
|_ A
|_ B
|_ C
|_ D
|   |_ E
```

Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

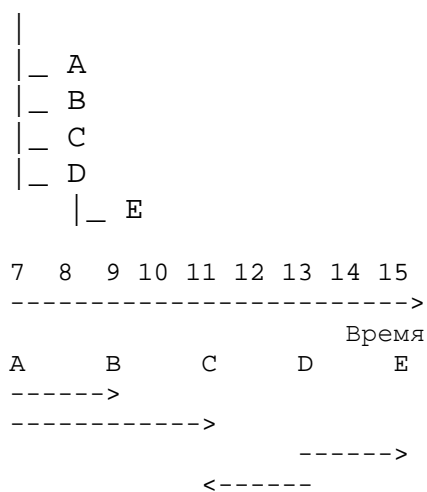
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 25



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

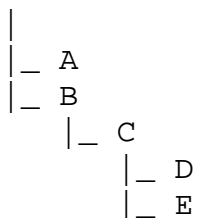
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 27



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

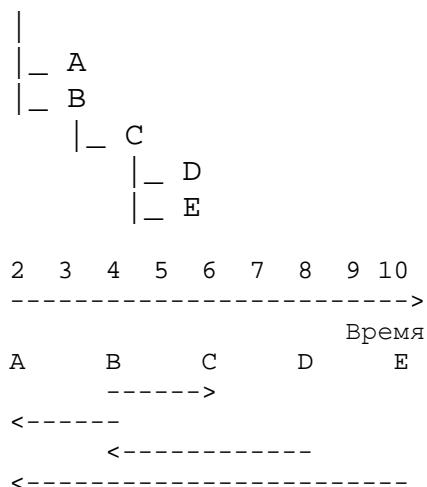
Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана OM и Поток событий некоторой ЭС. Построить ПМ OM ЭС.

Вариант 27





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

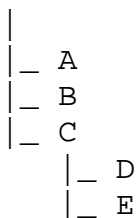
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 28



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента.
- Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

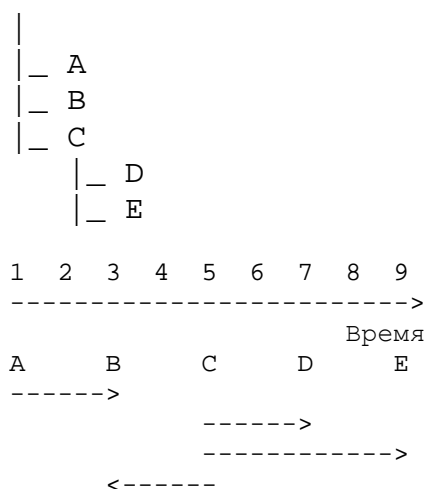
Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.

Вариант 28



### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

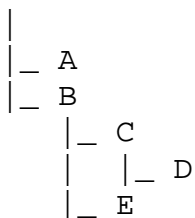
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 29



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWriteString свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

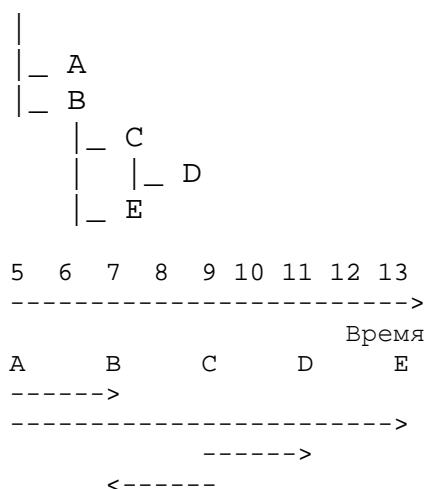
- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 29





### **Task\_01\_ReadWrite Property**

Создать класс A с ReadWrite свойством Name String типа, инициализация которого выполняется значением NameA при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA.

Отображаем значение свойства Name

Меняем свойство Name на новое значение

Отображаем значение свойства Name

### **Task\_02\_WriteOnceReadMany Property**

Создать класс A с ReadWrite свойством Name String типа и WriteOnceReadMany свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

Меняем свойство Salary на новое значение

Отображаем значение свойства Salary

### **Task\_03\_ReadOnly Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Меняем свойства Name и Salary на новые значения

Отображаем значение свойства Name и Salary

### **Task\_04\_Method**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary и задаем новое значение свойства Salary

Отображаем значение свойства Name и Salary

### **Task\_05\_WriteOnce Property**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA величину зарплаты 1000.

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

### **Task\_06\_ConstructorOverloads**

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnly String типа, закрытое поле, которого инициализируется при объявлении значением «пробела».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов и паролю увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

#### **Сценарий тестирования**

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_07\_MethodOverloads

Создать класс A с ReadOnly свойством Name String типа и ReadOnly свойством Salary Single типа, инициализация которых выполняется значениями NameA и 1000 соответственно при создании экземпляра класса.

Добавить свойство Password WriteOnce String типа, закрытое поле, которого инициализируется при создании экземпляра класса значением «пробел».

Перегрузить конструктор, который в первом случае инициализирует свойства Name и Salary значениями NameA и 1000 соответственно, а в другом случае - свойства Name, Salary и свойство Password значением пароля.

Добавить к классу метод RaiseSalary, который по заданному числу процентов увеличивает зарплату на это число процентов, а если повышение зарплаты превышает 10%, то запрашивает при этом еще и пароль.

Перегрузить метод RaiseSalary так, чтобы он по заданному числу процентов и заданному паролю, увеличивал зарплату и менее и более чем на 10%.

### Сценарий тестирования

Создаем объект класса A, передав конструктору класса значение имени объекта NameA и величину зарплаты 1000.

Создаем другой объект класса A, передав конструктору класса значение имени объекта NameA, величину зарплаты 1000 и пароль.

Отображаем значение свойства Name, Salary и пароль первого объекта

Отображаем значение свойства Name, Salary и пароль второго объекта

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

Вызываем для первого объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты менее 10%

Отображаем значение свойства Name и Salary

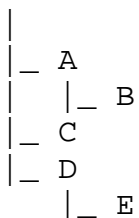
Вызываем для второго объекта метод RaiseSalary с уровнем повышения зарплаты более 10%

Отображаем значение свойства Name и Salary

## Task\_08\_Inherits

Дана ОМ некоторой ЭС.

Вариант 30



Пусть каждый класс ОМ имеет некоторый Public Main метод - MainИмяКласса.

Тестирование слова Protected

1) Пусть класс верхнего уровня иерархии имеет закрытое Protected ReadWrite String свойство ProtectedPropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ProtectedReadWritePropertyИмяКласса».

В Main методе производного класса самого нижнего уровня иерархии

- a) Отобразить значение свойства защищенного в классе самого верхнего уровня иерархии.
- b) Установить новое значение этого свойства величиной «ProtectedReadWritePropertyИмяПроизводногоКласса»
- c) Отобразить новое значение этого свойства
- d) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.

2) Пусть класс верхнего уровня иерархии имеет закрытое ReadWrite String свойство ReadWritePropertyИмяКласса, значение которого инициализируется при создании экземпляра класса величиной «ReadWritePropertyИмяКласса».

- a) В Main методе класса, находящегося вне иерархии, отобразить значение этого свойства.
- b) Установить новое значение этого свойства величиной «ReadWritePropertyИмяКлассаВнеИерархии»
- c) Отобразить новое значение этого свойства

Тестирование слова MyBase

3) Пусть базовый класс для класса самого нижнего уровня иерархии имеет метод Q, принимающий аргумент целого типа и возвращающий его удвоенную величину.

- a) В производном классе переопределить этот метод так чтобы он возвращал утроенную величину полученного аргумента. Вызвать переопределенный метод, а также вызвать соответствующий переопределяемый метод базового класса.

Замечание

- 1 Отображение значений свойств выполнить оператором MsgBox.
- 2 Тело Main метода любого класса начинать оператором MsgBox("Это Main метод класса ИмяКласса")

### Task\_09\_EventDynamic

Дана ОМ и Поток событий некоторой ЭС. Построить ПМ ОМ ЭС.  
Вариант 30

