

ВВЕДЕНИЕ

Целью Лабораторного практикума 3 является:

- освоение средств создания абстрактных классов;
- освоение средств реализации полиморфизма.

В результате выполнения Лабораторного практикума 3 студент должен

знать:

- способы и средства создания абстрактных классов;
- способы и средства реализации полиморфизма.

уметь:

- строить приложение, содержащее абстрактные классы и реализующие принцип полиморфизма;
- строить программную модель иерархии объектов;
- тестировать алгоритмы.

На примере из предметной области студенты получают первые навыки объектного анализа сложных систем, классификации объектов, определения состава свойств, методов и событий объекта.

Закрепляют средства создания объектов, свойств, методов, событий, задавать и использовать характеристики объектов.

Учатся создавать и использовать абстрактные классы и реализовывать принцип полиморфизма.

Осваивают технологии тестирования сложных систем.

Другой задачей, решаемой при выполнении заданий Лабораторного практикума 3, является освоение «Интегрированной среды разработчика приложений».

В процессе выполнения заданий студенты осваивают методы отладки алгоритмов, приемы визуального проектирования Windows приложений.

Построенные и протестированные проекты в дальнейшем будут использованы при разработке проектов – программных моделей экономических систем.

СОДЕРЖАНИЕ

1 ОФОРМЛЕНИЕ РАБОТЫ И УКАЗАНИЯ К ВЫПОЛНЕНИЮ

1.1 Оформление работы

Жесткая копия пояснительной записки включает Титульный лист и Лист заданий.

1.2 Указания к выполнению лабораторного практикума

Указания к «Литература»

Может иметь вид:

- 1 Браун С. Visual Basic 6: учебный курс – СПб: Питер, 2000.-576с.: ил.
- 2 Visual Basic 6.0: пер. с англ. – СПб: БХВ – Санкт-Петербург, 1999.- 992 с., ил.
- 3 Крейг Дж.К. и Уэбб Дж. Microsoft Visual Basic 6.0. Мастерская разработчика/Пер. с англ. – 5-е изд.- М.: Издательско-торговый дом «Русская редакция», 2001.- 720с.:ил.
- 4 Microsoft Corporation. Разработка приложений на Visual Basic 6.0. Учебный курс/Пер. с англ. – М.: Издательско-торговый дом «Русская редакция», 2000-400с.:ил.
- 5 Матросов В.А., Сергеев А.О., Чаунин А.П. HTML 4.0.- СПб.: Петербург, 2000.- 72с.:ил.
- 6 Палмер С. VBScript и ActiveX – СПб.: ЗАО «Издательство «Питер», 1999.- 384.: ил.
- 7 Уильям Мартинер Visual basic для Web: руководство разработчика: Пер. с англ. – К.: Издательская группа BHV, 2000.- 384 с.
- 8 МУ как делать встроенную систему помощи
- 9 МУ как делать вебсайты
- 10 МУ по ЛабПр_1
- 11 МУ по ЛабПр_2
- 12 МУ по ЛабПр_3
- 13 МУ по КурсРабоБПро

1.3 Порядок выполнения лабораторного практикума

Занятие 1

- 1 Получение заданий практикума
- 2 Построение фрагмента дерева каталогов
- 3 Подготовка шаблона файла отчета Report_LabPr3_Ek_02b_Sidorov.doc
- 4 Построение меню приложения

Занятие 2

- 1 Построение меню приложения
- 2 Построение палитры инструментов приложения

Занятие 3

Построение палитры инструментов приложения

Занятие 4 - 8

Повторение решения ЛабПр 2

Занятие 9 - 15

Решение ЛабПр 3

Занятие 16-18

Защита лабораторного практикума

2 НЕОБХОДИМЫЕ СВЕДЕНИЯ

2.1 Пример «AbstractClass»

Условие

Пусть объектный анализ некоторой ЭС приводит к следующей объектной модели ЭС

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

Пусть классы В и С имеют одинаковое по смыслу свойство «Код служащего», но имеющее различный формат.

Для класса В – это ВНомерСлужащего, а для класса С – Это СНомерСлужащего.

Пусть классы В и С имеет ReadOnly String типа свойство Имя (Name), иницируемое при создании экземпляра класса значениями NameВ и NameС соответственно.

Пусть класс А имеет MustOverride String типа свойство ID – КодСлужащего.

Задания

- 1 Создать объекты оВ и оС, иницируя свойство Name значениями NameВ и NameС соответственно.
- 2 Определить значение свойства ID объектов оВ и оС некоторым значением.
- 3 Отобразить значение свойства ID.

Указание 1

Объектный анализ задачи

Этот пункт имеет вид:

Пусть объектный анализ ЭС приводит к следующей ОМ ЭС

```
Employee (Employees)
|
|_ A
|_ B
    |_ C
    |_ D
        |_ E
```

Указание 2

Спецификации классов

Этот пункт имеет вид:

Класс А

Конструктор

Иницирует свойство Name

```
New(ByVal aName As String)
```

Свойства

NameA Имя объекта ReadOnly String типа

ID КодСлужащего MustOverride String типа

Методы

-

События

-

Указание 3

Наследование класса реализуют словом Inherits и далее пишут имя наследуемого класса

Указание 4

В заголовке абстрактного класса SystemX должно быть слово MustInherit.

Этот класс имеет конструктор для инициализации свойства Name

В этом классе реализовано ReadOnly свойство Name, а свойство ID не имеет тела и помечено словом MustOverride

В производных классах свойство ID должно быть реализовано и помечено словом Overrides

Указание 5

Тестирование может иметь вид:

Тест 1

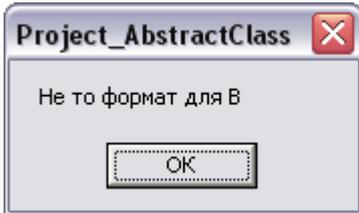
Если

oB.ID =5 oC.ID=C1

То ожидается ответ:

"Не то формат для B"

Действительно:



Тест 2

Если

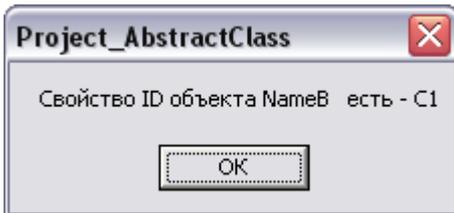
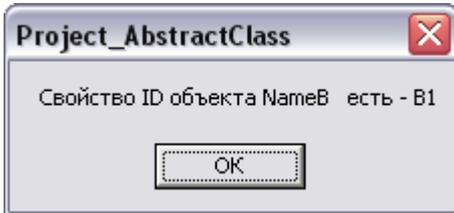
oB.ID =B1 oC.ID=C1

То ожидается ответ:

"Свойство ID объекта " & oB.Name & " есть - " & oB.ID)

"Свойство ID объекта " & oC.Name & " есть - " & oC.ID)

Действительно:

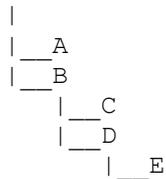


2.5 Пример «Polimorphizm»

Условие

Пусть объектный анализ некоторой ЭС приводит к следующей объектной модели ЭС

Employee (Employees)



Пусть класс Employee есть абстрактный класс. Имеет конструктор свойства Name. Реализует свойство Name. Имеет MustOverride ReadOnly свойство Salary Decimal типа. Имеет MustOverride метод RaiseSalary без параметров.

Пусть каждый класс (A,B,C,D,E) имеет ReadOnly String типа свойство Имя (Name), иницируемое при создании экземпляра класса значением NameИмяКласса.

Пусть каждый класс имеет ReadOnly Decimal типа свойство Зарплата, иницируемое при создании экземпляра класса некоторым значением.

Пусть каждый класс имеет перегруженный метод ПовышениеЗарплаты (RaiseSalary(ByVal ПроцентПовышенияЗарплаты as Byte)), изменяющий свойство Salary и реализующий некоторый алгоритм повышения зарплаты.

Пусть алгоритмы повышения зарплаты для всех классов различны.

Задания

- 1 Образовать экземпляры классов (объекты)
- 2 Поместить объекты в коллекцию Employees (на основе класса CollectionBase)
- 3 Итерируя по коллекции, отобразить значения свойств – ИмяОбъекта и Зарплата
- 4 Итерируя по коллекции, вызвать метод ПовышениеЗарплаты для каждого объекта и установить новую зарплату
- 5 Итерируя по коллекции, отобразить значения свойств – ИмяОбъекта и Зарплата

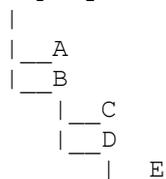
Указание 1

Объектный анализ задачи

Этот пункт имеет вид:

Пусть объектный анализ ЭС приводит к следующей ОМ ЭС

Employee (Employees)



Указание 2

Спецификации классов

Этот пункт имеет вид:

Класс А

Конструктор

Иницирует свойства Name и Salary

New(ByVal aName As String, ByVal aSalary As Decimal)

Свойства

NameA Имя объекта ReadOnly String типа

Salary Зарплата ReadOnly Decimal типа

Методы

RaiseSalary(ByVal PercentSalary as Byte) - повышение зарплаты

PercentSalary - входной параметр. Процент повышения зарплаты. Тип Byte. ОДЗ = [0-100].

События

-

Указание 3

Класс Employee

Абстрактный. Реализует свойство . Декларирует свойство Salary и метод RaiseSalary

Конструктор

Инициализирует свойство Name

New(ByVal aName As String)

Свойства

Public ReadOnly Property Name() As String

Public MustOverride ReadOnly Property Salary() As Decimal

Методы

Public MustOverride Sub RaiseSalary()

Повышение зарплаты

События

-

Класс Employees

Этот класс наследует пространство имен: System.Collections.CollectionBase

В этом классе достаточно реализовать свойство Item, возвращающее объект их коллекции по его индексу и метод Add, позволяющий добавить объект в коллекцию.

Свойства

Item(ByVal Index As Integer)

Объект коллекции.

Default, Public, ReadOnly, Object типа.

Index - входной параметр. Индекс элемента коллекции. Тип Integer.

ОДЗ =[0-ЧислоОбКоллекции -1]

Методы

Add(ByVal aObj As Object)

Добавить объект в коллекцию

aObj - входной параметр. Объект, добавляемый в коллекцию. Тип Object.

ОДЗ =[множество объектов]

События

-

Указание 4

Исходные коды классов

Сюда следует через Карман вставить исходные коды классов. Например:

Класс A

```
Public Class A
```

```
    Inherits Employee
```

```
    Private mSalary As Decimal
```

```
    Public Sub New(ByVal aName As String, ByVal aSalary As Decimal)
```

```
        MyBase.New(aName)
```

```
        mSalary = aSalary
```

```
    End Sub
```

```
Public Overrides ReadOnly Property Salary() As Decimal
    Get
        Return mSalary
    End Get
End Property
Public Overrides Sub RaiseSalary()
    mSalary = InputBox("Salary A = ", "", "200")
End Sub
End Class
```

Указание 5

Тестирование проекта

Тест 1

Если

oA.Name=NameA

oB.Name=NameB

oC.Name=NameC

oD.Name=NameD

oE.Name=NameE

oA.Salary=100

oB.Salary=100

oC.Salary=100

oD.Salary=100

oE.Salary=100

и

oA.Salary=200

oB.Salary=200

oC.Salary=200

oD.Salary=200

oE.Salary=200

То ожидается ответ:

Имя объекта = NameA Зарплата объекта = 100

Имя объекта = NameB Зарплата объекта = 100

Имя объекта = NameC Зарплата объекта = 100

Имя объекта = NameD Зарплата объекта = 100

Имя объекта = NameE Зарплата объекта = 100

Имя объекта = NameA Зарплата объекта = 200

Имя объекта = NameB Зарплата объекта = 200

Имя объекта = NameC Зарплата объекта = 200

Имя объекта = NameD Зарплата объекта = 200

Имя объекта = NameE Зарплата объекта = 200

Действительно:





Указание 6

Структура управляющего модуля проекта:

```
Sub Main()
```

Создаем объекты. Например

```
Dim oA As New A("NameA", 100)
```

и т.д.

Добавляем объекты в наборю Например

```
myEmployees.Add(oA)
```

и т.д.

Итерируя по набору отображаем свойства Имя и Зарплата

```
For Each en In myEmployees
```

```
MsgBox("Имя объекта = " & en.Name & " Зарплата объекта = " & en.Salary)
```

```
Next
```

Вызовем метод повышения зарплаты для каждого объекта

```
oA.RaiseSalary()
```

и т.д.

```
End Sub
```