

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНЖЕНЕРНОЙ МЕХАНИКИ И МАШИНОСТРОЕНИЯ

**КАФЕДРА «МЕХАНИЧЕСКОЕ ОБОРУДОВАНИЕ ЗАВОДОВ ЧЕРНОЙ МЕТАЛЛУРГИИ»
ИМ. ПРОФ. СЕДУША В.Я.**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению практических занятий

по дисциплине вариативной части

по выбору студента профессионального цикла

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

РЕМОНТНОГО ПРОИЗВОДСТВА

для студентов всех форм обучения

направления подготовки 15.03.02

«Технологические машины и оборудование»

**ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНЖЕНЕРНОЙ МЕХАНИКИ И МАШИНОСТРОЕНИЯ

**КАФЕДРА «МЕХАНИЧЕСКОЕ ОБОРУДОВАНИЕ ЗАВОДОВ ЧЕРНОЙ МЕТАЛЛУРГИИ»
ИМ. ПРОФ. СЕДУША В.Я.**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к проведению практических занятий

по дисциплине вариативной части

по выбору студента профессионального цикла

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

РЕМОНТНОГО ПРОИЗВОДСТВА

для студентов всех форм обучения

направления подготовки 15.03.02

«Технологические машины и оборудование»

Рассмотрены на заседании
кафедры «Механическое оборудование
заводов черной металлургии»
им. проф. Седуша В.Я.
Протокол № 11 от 03.04.2017 г.

Утверждены на заседании
учебно-издательского совета ДОННТУ
Протокол № ___ от __.__. 20__ г.

Донецк
ДОННТУ
2017

УДК 53.083

Методические указания к проведению практических занятий по дисциплине вариативной части по выбору студента профессионального цикла «Информационное обеспечение ремонтного производства» для студентов всех форм обучения направления подготовки 15.03.02 «Технологические машины и оборудование» / сост.: Н. А. Ченцов, В. А. Сидоров, С. Л. Сулейманов. – Донецк: ГОУВПО «Донецкий национальный технический университет», 2017. – 49 с.

Отражены цели и задачи практических занятий по дисциплине «Информационное обеспечение ремонтного производства» для студентов всех форм обучения по направлению подготовки 15.03.02 «Технологические машины и оборудование». Описаны структура занятий, порядок подготовки к ним, последовательность операций и действий, направленных на выполнение поставленных задач, даны рекомендации по использованию теоретического материала.

Составители: Ченцов Н. А., д.т.н., доцент, профессор кафедры «Основы проектирования машин», Сидоров В. А., д.т.н., доцент, профессор кафедры «Механическое оборудование заводов черной металлургии» им. проф. Седуша В.Я., Сулейманов С. Л., ассистент кафедры «Мехатронные системы машиностроительного оборудования».

Рецензенты: д.т.н., профессор А.П. Кононенко
д.т.н., профессор С.П. Еронько

Ответственный за выпуск:

к. т. н., доцент Е. В. Ошовская

© А. Л. Сотников

СОДЕРЖАНИЕ

1. «С» СТУДИЯ NETBEANS ДЛЯ JAVA	5
С0 Инсталляция студии <i>NETBEANS</i>	5
С1 ЛР 11. Студия <i>NETBEANS</i>	6
С2 ЛР 12 «Компоненты проекта»	7
Индивидуальное задание (z1) 11 «Студия»	8
2. «И» ИНЖЕНЕРНЫЕ ЗАДАЧИ	10
и1 ЛР 21 «Создание проекта (приложения)» <i>RI1CMMSFIO</i>	10
и2 ЛР 22 «Изменение свойств компоненты на форме» <i>RI1CMMSFIO</i>	12
z2 (21) «Конструктор»	13
и3 ЛР 23 «Создание события в кнопке» <i>RI3FIO</i>	13
z3_ «Калькулятор» ИЗ «Событие в кнопке»	15
и4 ЛР 24 «Создание запускаемой дополнительной формы» <i>RI3FIO</i>	15
и5 ЛР 25 «Создание формы с меню» <i>RI3FIO</i>	16
z4_ ИЗ 23 «Меню»	17
3. СЕРВЕР БАЗЫ ДАННЫХ	18
Б1 ЛР 31 «Теоретические основы баз данных»	18
z5 «Таблица» ИЗ 31 «Формализовать таблицу»	19
Б2 ЛР 32 «Запуск СУБД <i>JAVA DB</i> » <i>DB SAMPLE</i>	19
Б3 ЛР 33 «Создание БД» <i>DB DBFIO</i>	21
Б4 ЛР 34 «Создание записей в таблице» <i>DB «DBFIO»</i>	23
z6_ ИЗ 32 «Создание БД»	24
4. ПРИЛОЖЕНИЕ ДЛЯ РАБОТЫ С БД	25
П1 ЛР 41 «Создать приложение «Перечень друзей»» <i>RP1FRIENDFIO</i>	25
П2 ЛР 42 «Работа в приложении «Перечень друзей»» <i>RP1FRIENDFIO</i>	27
z7 «Диагностика» ИЗ 41 «Создать приложение «Диагностика»»	28
П3 ЛР 43 «Чтение поля из таблицы базы данных» <i>RP3FIOREAD</i>	29
П4 ЛР 44 «Знакомство с циклом FOR» <i>RP3FIOREAD</i>	30
П5 ЛР 45 «Знакомство с условием IF» <i>RP3FIOREAD</i>	32
z8 «Средняя V» __ ИЗ 42 «Средняя скорость изменения Y»	34
5. ЧИСЛЕННЫЙ МЕТОД РЕШЕНИЯ ЗАДАЧИ	35
Ч1 ЛР 51 «Использование численного метода»	35
Ч2 ЛР 52 «Поиск экстремума»	37
Ч3 ЛР 52 «Знакомство с циклом WHILE» <i>RN3FIOWHILE</i>	38
z9 «WHILE» ИЗ 51 «Оценка точности решения задачи»	40
6. ДОКУМЕНТАЦИЯ К РАЗРАБОТКЕ ПРИЛОЖЕНИЯ	41
Д1 ЛР61. Разработка блок – схемы решения задачи	41
z10 «Блок – схема» ИЗ 61 «Блок – схема к определению Totk»	42
z11 «Приложение Totk» ИЗ 62 «Определение даты отказа»	43
Д2 ЛР62. Комплект документов к решению задачи	44

7. ЗАЧЕТНАЯ РАБОТА

48

ZR “РАЗРАБОТАТЬ КОМПЛЕКТ ДОКУМЕНТОВ К ОПРЕДЕЛЕНИЮ ЭКСТРЕМУМА ФУНКЦИИ” 48

1. «с» Студия NetBeans для JAVA

СО Инсталляция студии NetBeans

- 1) Исходные материалы - у преподавателя получить архивированный файл *21СучТехнолТО ч1.zip*.
- 2) Исходные материалы - у преподавателя получить инсталляционный файл:
 - *jdk-8u20-nb-8_0_1-windows-x64.exe* (для 7ки и выше)
 - или *jdk-8u20-nb-8_0_1-windows-i586.exe* (для XP).
- 3) Для установки программного обеспечения необходим компьютер под управлением операционной системы *Windows XP* или старше.
- 4) Подготовка ПК студента к инсталляции:
 - на ПК студента создать папку *javaFIO*, где *FIO* заменить на **инициалы студента**;
 - в папку (*javaFIO*) переписать заархивированный файл и разархивировать его – будет создана папка *javaFIO / 21СучТехнолТО ч1*;
 - папку *javaFIO / 21СучТехнолТО ч1 / CMMSprimer* скопировать непосредственно в папку студента – должно получиться *javaFIO / CMMSprimer*;
 - в папке *javaFIO / 21СучТехнолТО ч1 / KTVRS install J* открыть инсталляционные файлы:
 - *jdk-8u20-nb-8_0_1-windows-i586.exe* (для XP);
 - *jdk-8u20-nb-8_0_1-windows-x64.exe* (для 7ки и выше).
- 5) Инсталляция языка *Java* и студии *NetBeansIDE*– запустить необходимый инсталляционный файл в папке *javaFIO / 21СучТехнолТО ч1 / KTVRS install J* и следовать инструкциям;



*На столе появится ярлык - Запустить приложение, сделать *wClick* на ярлыке. Откроется студия *NetBeans*.

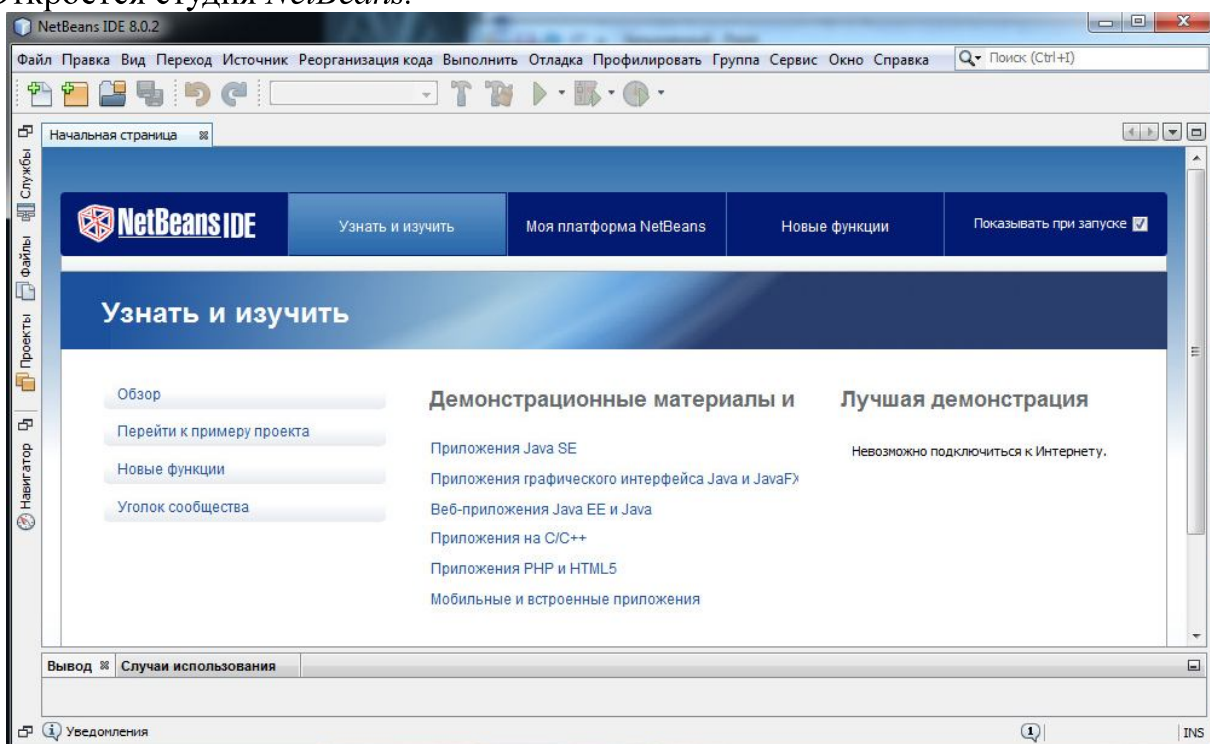


Рисунок - Студия NetBeans для языка программирования JAVA

с1 ЛР 11. Студия NetBeans

Цель: знакомство со студией *NetBeans*.

Исходные условия: наличие в ПК студента папки *D:\javaFIO \ CMMSprimer* с примером проекта.

Постановка задачи: освоить: запуск и остановку студии *NetBeans*, ознакомиться с ее компонентами. Выполнить загрузку, запуск и остановку готового проекта.

Выполнение работы

Запуск

- 1) Запустить студию *NetBeans* – *Пуск/Программы/NetBeans/NetBeans IDE* *Будет загружена студия с начальной страницей *NetBeans*.
- 2) Загрузить пример проекта – выполняется студии в следующей последовательности:
 - В главном меню выбрать *Файл\ Открыть проект* \ найти свою папку.
 - Выбрать папку с примером проекта - *wClick* на *CMMSprimer*. *Загрузится проект, имя которого, *CMMS* видно в левом верхнем углу студии на панели *Проекты*.
 - Загрузить стартовую форму – в студии на левой панели *Проекты* открыть *CMMS \ Пакеты исходных файлов \ CMMS \ startCMMS.java*, сделать на файле *rClick* и выбрать *Открыть*. *В центре студии будет создана вкладка, на которую загрузится компоненты формы, описанной в файле.
- 3) Загрузить палитру с графическими компонентами – слева нажать вкладку «Палитра». Откроется панель с компонентами палитры используемыми при разработке программы.

Компоненты студии

- 4) Студия включает пять основных компонент, рисунок 1.1.

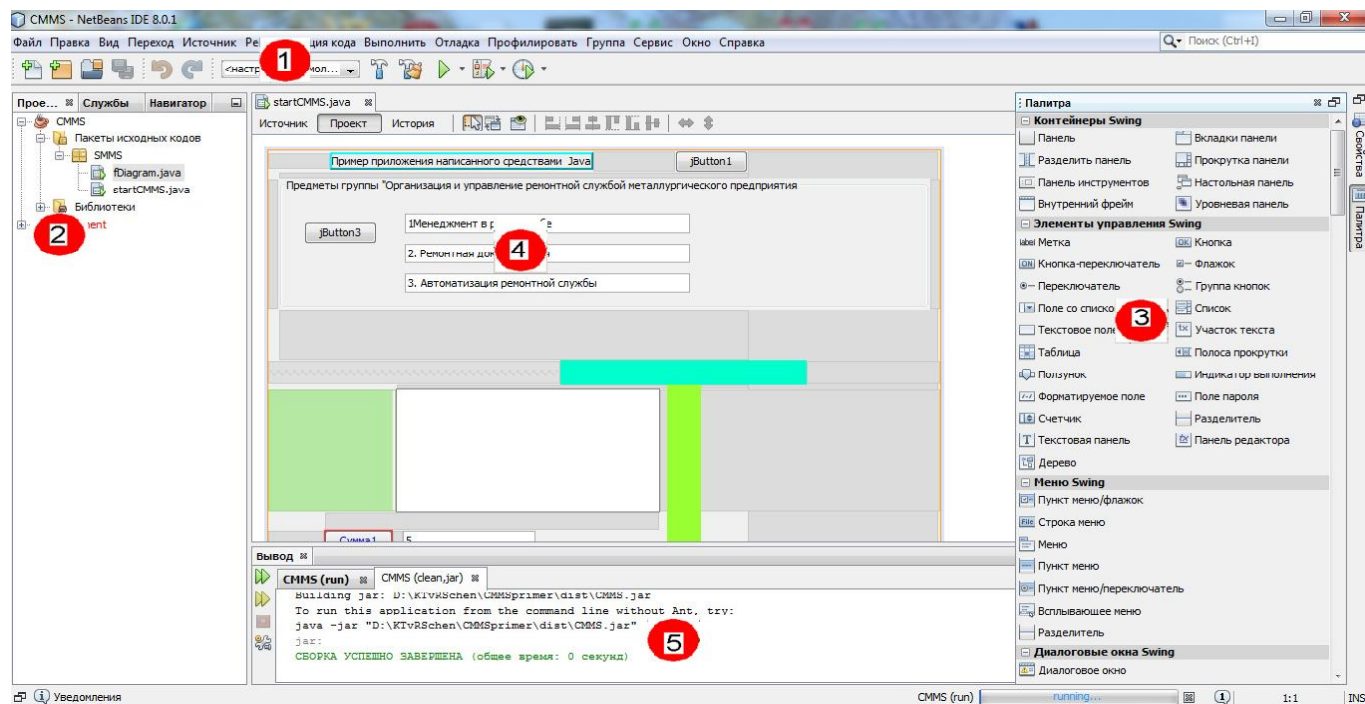


Рисунок 1.1 Компоненты студии *NetBeans*

Характеристика компонент:

- 1. *Главное меню* (верхняя панель);
- 2. *Навигатор компонент* проекта (левая панель);
- 3. *Палитра* графических компонентов (правая панель)
- 4. *Редактор класса* - для корректировки кода и графического представления (центральная панель).
- 5. *Отчет* о выполнении компиляции проекта (нижняя панель).

Работа с приложением

- 5) Запуск приложение (запустить проект на выполнение) – в главном меню нажать *Выполнить / Запустить проект*. *Приложение запустится и в левом верхнем углу экрана появится панель «*Пример приложения написанного средствами Java.*».
- 6) Остановить приложение – *Click* на «X» в правом верхнем углу запущенной формы.
- 7) Закрыть проект в студии – *Файл \ Закрыть проект (CMMS)*.
- 8) Закрыть студию – *Файл \ Выход*

После выполнения показать преподавателю:

с2 ЛР 12 «Компоненты проекта»

Цель: ознакомиться с компонентами проекта.

Исходные условия: наличие в ПК папки *D:\javaFIO \ CMMSprimer* с примером проекта.

Постановка задачи: найти компоненты проекта в среде студии *NetBeans* и файловой системе ПК. Исследовать их содержание и место в проекте.

Выполнение работы

Запуск

- 1) Запустить студию.
- 2) Открыть проект *CMMSprimer* (приложение) и загрузить его стартовый класс *CMMS \ Пакеты исходных файлов \ CMMS \ startCMMS.java*, сделать на файле *rClick* и выбрать *Открыть*.
- 3) На вкладке *Проекты* (см. рисунок 2.1.) приведены компоненты, с которыми непосредственно работает кодер во время разработки программного обеспечения в студии. Они представлены деревом в начале которого находится наименование проекта *CMMSprim* и его папки:
 - *Пакеты исходных файлов* содержит папки с именами пакетов, например, *pFio*. Каждый пакет содержит файлы проекта, один из которых является главным классом (запускаемым), например, *pFio / startCMMS.java*.
 - Библиотеки – содержат стандартные классы.
- 4) Один пакет (папка в *Пакеты исходных файлов*) может включать несколько файлов, каждый из которых является классом. В студии *NetBeans* содержание класса выводится в *Редакторе класса* (центр студии) на две панели:
 - *Источник* – где обеспечивается возможность непосредственной корректировки текста класса (например, написания текста событий).
 - *Проект* (конструктор) – где показано графическое представление класса (формы).

Компоненты проекта в файловой системе компьютера.

- 5) Выйти из студии – закрыть ее.
- 6) В файловой системе найти свою папку *D:\ javaFIO \ CMMSprimer* с примером проекта.
- 7) В папке *CMMSprimer* найти интерес представляют папки проекта:
 - *Dist-* содержит два файла, в том числе с расширением *.jar* (запускаемое приложение *CMMS.jar*). Приложение может запускаться из файловой системы ПК, где установлена виртуальная *java* машина.
 - *Lib* - содержат стандартные классы (библиотека).
 - *Src* – содержит формы (классы) проекта. *Доступ к ним так же возможен с вкладки *Проекты*

Запуск приложения из файловой системы

- 8) В файловой системе найти файл *D:\ javaFIO \ CMMSprimer \ dist\ CMMS.jar*.
- 9) Запустить файл (приложение) на выполнение - сделать *wClick*.
Закрыть приложение.
После выполнения показать преподавателю;

Индивидуальное задание (z1) 11 «Студия»

1. **Создать текстовый** файл *z1fio.doc* в своей папке. **В именах FIO заменить на инициалы студента. Размер окончательного файла z1fio.doc - до 500 кБайт.**

Далее выполняется в загруженном проекте *CMMSprimer*.

2. На левой панели *Навигатор компонент проекта* открыть вкладку «Проекты» раскрыть содержание проекта *СММС*. На центральной панели *Редактор класса* - открыть конструктор проекта (вкладка *Проект*) (см. рисунок 1).

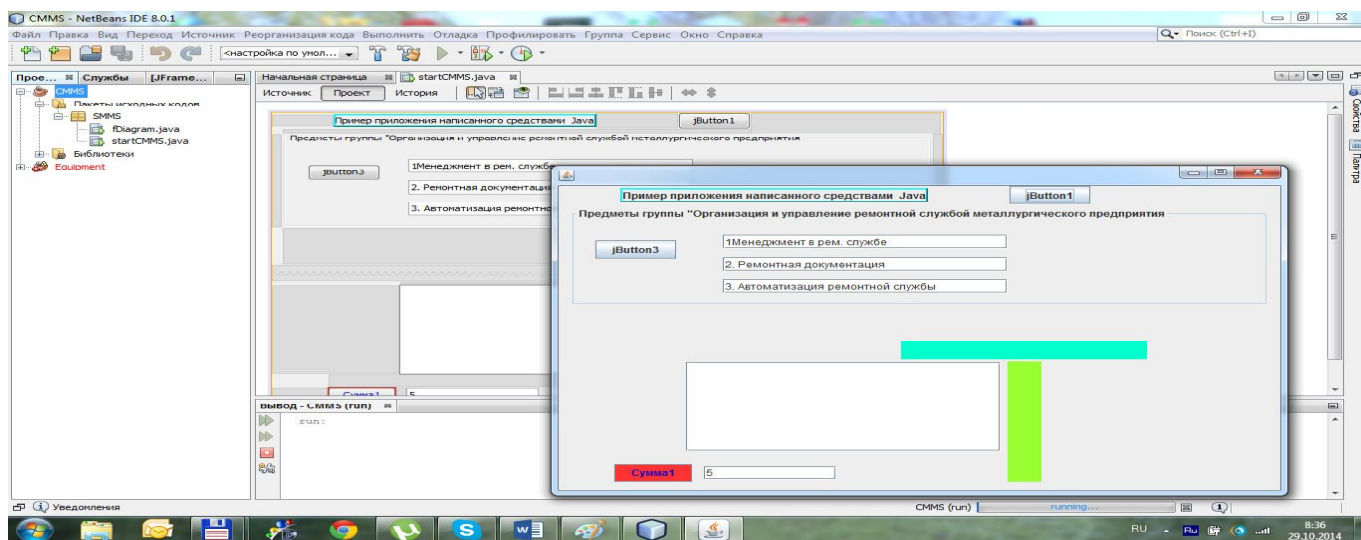


Рисунок 1. Компоненты проекта и конструктора

3. Запустить проект *СММС* (в главном меню нажать *Выполнить / Запустить проект*).
4. На экране скомпоновать компоненты студии NetBeans и запущенного приложения в соответствии с рисунком 1.
5. **Сделать копию экрана** (кнопка *Print Screen* на клавиатуре), загрузить в графический редактор, например, *Paint*, обрезать все лишнее, сделать «Сохранить как» и выбрать тип файла *.JPEG*.
6. Полученный графический файл вставить в текстовый файл *z1fio.doc* с именем «Рисунок 1. Компоненты проекта и конструктора».
7. На панели *Навигатор компонент проекта* (левая панель) выбрать вкладку *Файлы* и раскрыть папку *src / rFio*. На центральной панели *Редактор класса* - открыть текст программы (вкладка *Источник*)
8. **Сделать копию экрана** обработать и сохранить в файле *z1fio.doc* с именем «Рисунок 2. Файлы проекта в студии и текст класса *startCMMS.java*».
9. Закрыть студию и выйти в файловую систему. В файловой системе раскрыть папку *???:\javaFIO \CMMSprimer\src\SMMS * - файлы с текстами форм проекта *CMMSPrimer*. Открыть папку *???:\javaFIO \CMMSprimer\dist* - исполняемый файл *CMMS.jar* запускаемый из файловой системы.
10. Из файловой системы запустить исполняемый файл разместить компоненты экрана подобно рисунку 2.

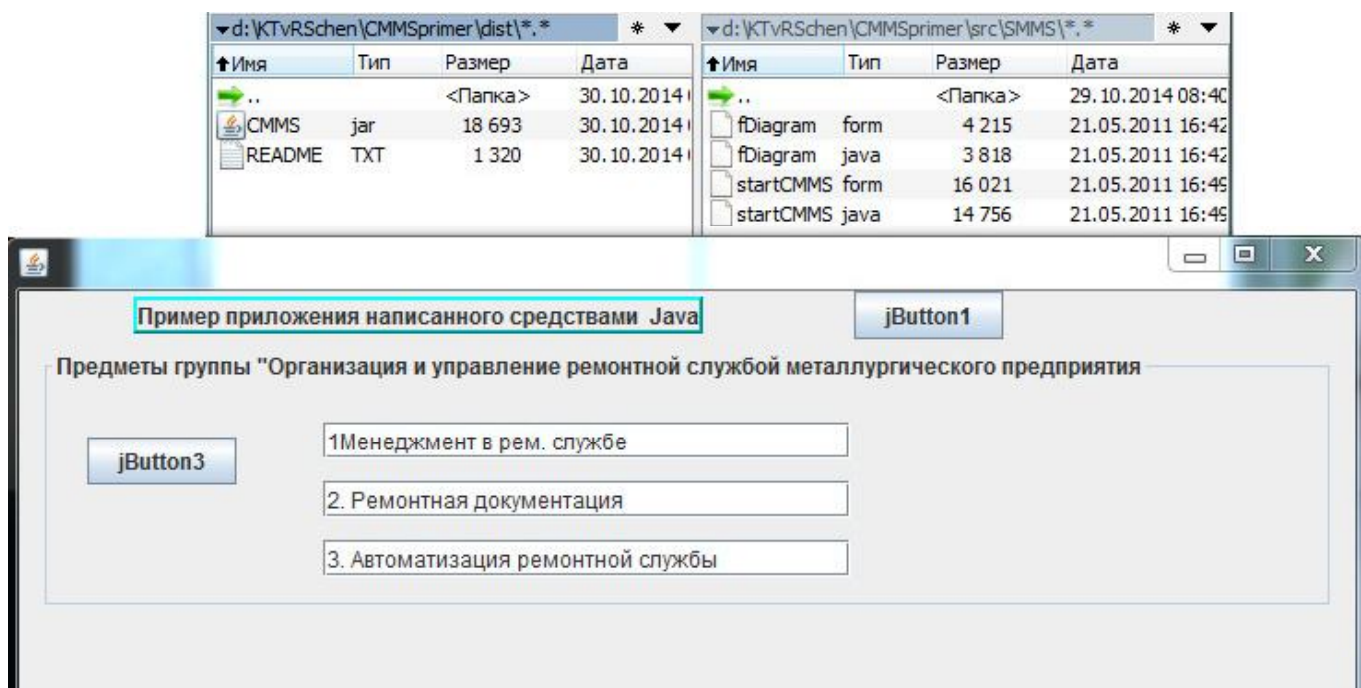


Рисунок 2. Файлы проекта в файловой системе и запущенное приложение

11. Сделать копию экрана, обработать ее и вставить в текстовый файл *z1fio.doc* с именем «Рисунок 3. Файлы проекта в файловой системе и запущенное приложение»

12. Файл *z1fio.doc* уменьшить до размера 500кБайт и переслать на *dist.donntu.org* или передать преподавателю.

Пример *z1* выполненного Ченцовым Н.А. приведен в файле *iz11 ОБРАЗЕЦ.docx* (335кбайт).

2. «и» ИНЖЕНЕРНЫЕ ЗАДАЧИ

Везде FIO заменять на инициалы студента, например, у Ченцова это chen (CMMSchen)

и1 ЛР 21 «Создание проекта (приложения)» *ri1CMMSfio*

Цель: получить первичные навыки решения задач в студии– создание проекта (приложения) и формы (класса).

Исходные условия: наличие в ПК папки *D:/javaFIO*.

Постановка задачи: создать простейший проект обеспечивающий вывод на экран заданной надписи.

Выполнение работы

1) Запустить студию NetBeans – Пуск/Программы/NetBeans/NetBeans IDE.

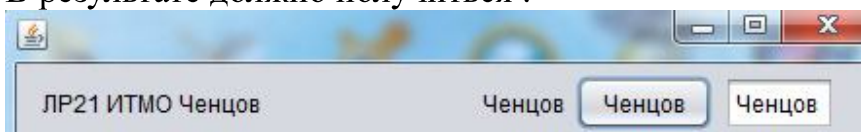
2) Создание проекта:

- В меню студии выбрать *Файл > Создать проект* – будет открыта форма «Создать проект» на ней выполнить.
- На панели "*Категории*" выберите *Java*;
- На панели «*Проекты*» выберите *Приложение Java*.
- Нажмите кнопку *Далее*.

- На форме *«Новый Приложение JavaСМMS»*.
 - В поле *«Имя проекта»* укажите *ri1СМMSfio*.
 - В поле *«Расположение проекта»* укажите свою папку *D:/javaFIO* в файловой системе.
 - Не выбирать *«Использовать отдельную папку для библиотек»*.
 - Не выбирать *«Создать главный класс»*.
 - Нажмите кнопку *Готово*. *Будет создан проект, и дерево его компонентов станет видимым на левой панели (вкладка *Проекты*).
- 3) Создание стартовой формы (класса).
- На наименовании проекта *ri1СМMSfio* сделать *rClick*, затем выбрать *Новый > Форма JFrame*. *Откроется панель *New Форма JFrame*.
 - В поле *Имя класса* укажите *ingFio*;
 - В поле *Пакет* укажите *pСМMS*;
 - Нажмите кнопку *Готово*.
- *В проекте будет показана ветвь пакета *pСМMS* и ее компонента – класс *ingFio.java*.
- *В центральном окне откроется вкладка *ingFio.java* с выбранным конструктором (панель *Проект*).
- 4) Создание компоненты (надписи) на форме.
- Закрыть все вкладки в *Редакторе класса* (центральное окно).
 - Загрузить стартовый класс на редактирование – на панели *Проекты* указать файл *ingFio.java* сделать на нем *rClick*, затем – *Открыть*. *В Редакторе класса будет создана вкладка *ingFio.java*.
 - Открыть конструктор формы – нажать вкладку *«Проект»*. *Откроется графическое представление формы (пока без компонентов).
 - Выбор новой компоненты для размещения - на правой панели *«Палитра»* найти компоненту *«Метка»* и сделать на ней *Click*.
 - Размещение компоненты на форме – сделать *Click* на форме – появится надпись *jLabel1*.
 - Вызвать содержание надписи на редактирование– сделать на ней *wClick* (она посинеет – перейдет в режим редактирования).
 - Выполнить редактирование - написать *ЛР21 ИТМО ФИО* и нажать *Enter*.
- * Останется новое содержание надписи.
- 5) Запустить проект на выполнение
- В главном меню нажать *Выполнить > Запустить проект*. *Откроется панель *Выполнить проект*.
 - На панели *Выберите главный класс* указать *pСМMS.ingFio* (он будет синим) нажать *ОК*. В правом нижнем углу будет двигаться зеленый курсор – идет обработка данных.
 - После его остановки запустится приложение и в левом верхнем углу экрана откроется созданная форма с надписью *ЛР21 ИТМО ФИО*.
 - Закрыть (остановить) приложения – в главном меню нажать *Выполнить \ Остановить сборку/выполнение...* (нижняя строка).

Дополнительное задание.

- Добавить на форму из палитры: метку, кнопку, текстовое поле. Сгруппировать их в одну строку и в каждой из них записать свою фамилию. Запустить приложение и остановить его.
- В результате должно получиться :



• После выполнения показать преподавателю.

и2 ЛР 22 «Изменение свойств компоненты на форме» *ri1CMMSfio*

Цель: получить первичные навыки корректировки свойств компонент.

Исходные условия: наличие проекта *ri1CMMSfio* с классом *ingFio*.

Постановка задачи: выполнить корректировку свойств компонент «Текстовое поле», «Рамка».

Выполнение работы

- 1) Запустить студию и загрузить класс *ri1CMMSfio \ pCMMS \ ingFio.java*.
- 2) Создать надпись *ЛР22 ИТМО ФИО*.
- 3) Создать текстовое поле на форме – взять с *Палитры* компоненту *Текстовое поле* и перенести на форму.
- 4) Открыть перечень свойства компоненты *Текстовое поле* - на ней *rClick > Свойства* откроется перечень ее свойств:

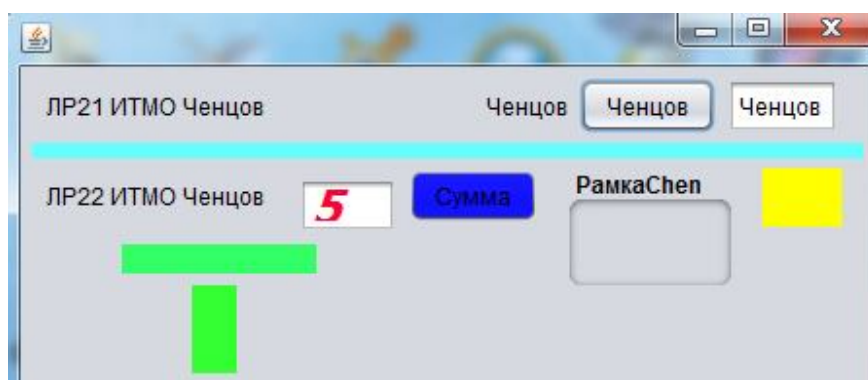
Background	Цвет фона.
Font	Шрифт, которым делается надпись на компоненте.
Foreground	Цвет шрифта, которым делается надпись на компоненте.
Text	Текст (надпись) на компоненте.
Прочие	

- 5) Изменить в компоненте:
 - надпись на ней - в *Свойствах* присвоить – *text = 5*.
 - ширину компоненты - в *Свойствах* выбрать параметр *preferredSize* и нажать три точки. На панели в поле *Ширина* указать *50*.
 - шрифт надписи - в свойствах изменить: стиль шрифта на *полужирный курсив*; размер – *24*; цвет текста (*foreground*) – *красный*;
 - имя компоненты - на ней *rClick*, выбрать *Изменить имя переменной* и присвоить *Новое имя = fioResalt*. *Увеличить высоту компоненты.
- 6) Создать кнопку на форме – использовать компоненту *Кнопка* из *Палитры*. *Свойства
- 7) Изменить надпись на кнопке - в *Свойствах* присвоить – *text = Сумма*.
- 8) Изменить цвет кнопки - в *Свойствах* изменить *background*.

- 9) Изменить имя компоненты - на ней *rClick*, выбрать *Изменить имя переменной* и присвоить *Новое имя = fioSuma*.
- 10) Создать панель и изменить ее цвет Палитра > Панель > Свойства background = выбрать цвет.
- 11) Создать рамку с надписью Палитра > Панель > Свойства > border > Рамка с надписью > Заголовок = "Рамкаfio".
- 12) Запустить приложение проверить свойства созданных компонент.

Дополнительное задание.

- Используя компоненту *Панель* отделить компоненты *ЛР21* от *ЛР 22*.
- Используя две компоненты *Панель* на форме создать синюю букву Т.
В результате должно получиться



После выполнения показать преподавателю, в том числе имена текстового поля и кнопки.

z2 (21) «Конструктор»

Постановка задачи: создать приложение содержащее заданные компоненты. Результаты выполнения инд. задания разместить в файле *z2fio.doc*.

Выполнение.

1. Создать проект *z2fio*, пакет *p2fio* и стартовую форму в нем *start2fio*.
- 2.Используя компоненту "метка" в левом верхнем углу формы указать код индивидуального задания - *z2fio*.
- 3.Используя компоненту "метка" создать на форме дату своего дня рождения, например, *28.09.1993*. В ней изменить характеристики шрифта.
- 4 Создать компоненту "текстовое поле" написать в ней «Инд.задание №2 – Выполнил студент группы *FIO*» и присвоить ей имя *nameFio*.
- 5.Используя компоненты "панель" создать красную букву Ш
6. Запустить приложение, наложить его на конструктор, подобно рисунку 1 (стр.9).
- 7.Сделать копию экрана **вставить** ее в созданный текстовый файл *z2fio.doc*. Файл переслать преподавателю на *dist*.

и3 ЛР 23 «Создание события в кнопке» *ri3Fio*

Цель: научиться создавать события, обеспечивающие решение простейшей инженерной задачи – суммирование двух чисел.

Исходные условия: наличие в ПК папки *D:/javaFIO*.

Постановка задачи: разработать программу обеспечивающую чтение с экрана заданного числового значения, увеличение его на величину константы и последующий вывод на экран.

Выполнение работы

- 1) Запустить студию. В папке *javaFIO* Создать проект *ri3Fio* с пакетом *r3Fio* и стартовым классом *ing3Fio.java*.
- 2) Загрузить *ing3Fio.java* и открыть *конструктор класса* - активировать панель *Проект* на вкладке *ing3Fio.java*
- 3) Создать текстовое поле на форме с именем *fioResalt* и надписью 5
- 4) Создать кнопку на форме с именем *fioSuma* и надписью *Сумма*.
- 5) Создать событие в кнопке *fioSuma*
 - Открыть свойства - на кнопке сделать *rClick*, выбрать *Свойства*. *Откроется панель *свойства*.
 - Перейти в события - в верхнем меню панели выбрать *События*,
 - Создать событие (нажатие кнопки мышкой - *mouseClicked*) – найти параметр *mouseClicked* и нажать треугольник в ее строке – будет предложено создать *fioSummaMouseClicked* –подтвердить создание (нажать наименование). *В строке параметра появится эта надпись – имя события, наступающего в приложении при нажатии мышкой на кнопку.
 - Закрыть панель *Свойства*.
- 6) Загрузка кода события в кнопке *fioSuma*:
 - Открыть конструктор (вкладка *Проект*).
 - На кнопке открыть *Свойства* и в *События* указать *fioSummaMouseClicked*
 - На нем сделать *Click*.
 - Закрыть свойства – нажать кнопку *Закрыть*. *Курсор перейдет на панель *Источник* и будет стоять в первой строке метода (процедуры) соответствующей событию *fioSummaMouseClicked*.
- 7) Содержание кода события *fioSummaMouseClicked*. *В событии разместить код (подчеркнутые строки) – в результате должно получиться следующее.


```
private void fioSummaMouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
//внутренняя переменная
float fioTemp;
//чтение данных с формы – типа float
fioTemp = Float.parseFloat(fioRezalt.getText());
//выполнение расчета
fioTemp = fioTemp +7;
//передать результат на форму
fioRezalt.setText(String.valueOf(fioTemp));
}
```
- 8) Текст события можно скопировать из методички и вставить в событие (в программу).

- 9) Ошибка – **fioRezalt** будет подчеркнута красным. Исправить **fioRezalt** на **fioResalt** - имя текстовой компоненты на форме (в которой указано «5»).
- 10) Запустить приложение на выполнение и проверить суммирование. Остановить приложение.

Дополнительное задание.

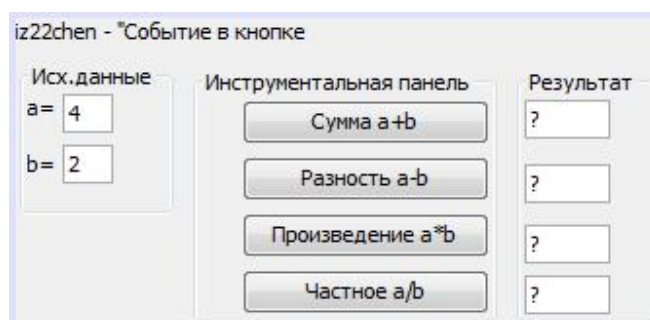
- Добавить на форму текстовое поле.
 - Обеспечить вывод в поле **fioResult** суммы его значения и значения из нового текстового поля.
- После выполнения показать преподавателю.

z3 «Калькулятор» ИЗ«Событие в кнопке»

Постановка задачи: создать приложение обеспечивающее выполнение простейших арифметических операций (аналог калькулятора). Результаты выполнения инд. задания разместить в файле *z3fioCalk.doc*.

Выполнение.

- 1) Создать проект *z3fio*.
- 2) Создать пакет *pz3fio* и стартовую форму *fz3fio*.
- 3) Создать экранную форму с компонентами в рамках:
 - Исходные данные (*Текстовое поле*): *a* (значение – номер в списке группы), *b* (значение 2);
 - Инструментальная панель с кнопками (*Кнопки*): Сумма $a+b$; Разность $a-b$; Произведение $a*b$; Частное a/b ;
 - Результат - четыре текстовых поля (*Текстовое поле*) для вывода результатов вычислений.



- 4) В кнопках создать события, обеспечивающие выполнение математических операций, написанных на них.
- 5) Запустить приложение, наложить его на конструктор, подобно рисунку 1 (стр.9).
- 6) Сделать копию экрана **вставить** ее в созданный текстовый файл *iz22fio.doc*.
- 7) В текстовый файл *z3fioCalk.doc*. **добавить коды событий** в кнопках.
- 8) Файл переслать преподавателю на *dist*.
 - *Оформление файла *z3fioCalk.doc* смотри в его примере *iz22fio.doc*.

и4 ЛР 24 «Создание запускаемой дополнительной формы» *ri3Fio*

Цель: научиться создавать дополнительную форму, запускаемую из стартовой.

Исходные условия: наличие проекта *ri3Fio* с главным (стартовым) классом *ing3Fio.java*.

Постановка задачи: в стартовом классе создать кнопку и код в ней обеспечивающих запуск дополнительного класса.

Выполнение работы

- 1) Загрузить проект *ri3Fio*.
- 2) Создать дополнительную форму (класс).
 - На наименовании пакета *ri3Fio* сделать *rClick*, выбрать *Новый*, затем *Форма JFrame*
 - В поле «Имя класса» укажите *fioDiagram*;
 - В поле «Пакет» выберите *p3Fio*;
 - *Готово*.
- *В ветви пакета *p3Fio* появится класс *fioDiagram* а в окне редактора откроется вкладка *fioDiagram.java* с открытым конструктором формы (*Проект*).
- 3) Создать метку на форме и в ней написать **ФОРМА ДЛЯ ПОСТРОЕНИЯ ДИАГРАММ fio**.
- 4) Загрузить стартовый класс *ing3Fio* на редактирование.
- 5) Создать в классе *ing3Fio* кнопку с именем *fioDia*, надписью *Диаграмма* и событием *fioDiaMouseClicked*.
- 6) Отредактировать код события *fioDiagramMouseClicked* - должно получиться следующее.


```
private void fioDiaMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    new fioDiagram().setVisible(true);
}
```
- 7) Запустить приложение на выполнение и проверить загрузку дополнительной формы при нажатии кнопки *fioDia*. Остановить приложение.

Дополнительное задание.

- Создать форму «Тест» с надписью *Тестfio*.
- Для ее запуска на форме *fioDiagram* создать кнопку.
- Запустить приложение и проверить работу.

После выполнения показать преподавателю.

и5 ЛР 25 «Создание формы с меню» *ri3Fio*

Цель: научиться создавать типовой набор взаимосвязанных форм, используемый в приложениях.

Исходные условия: наличие проекта *ri3Fio* с главным (стартовым) классом *ing3Fio.java*.

Постановка задачи: создать форму МЕНЮ обеспечивающую запуск любой формы приложения.

Выполнение работы

- 1) Загрузить студию и проект *ri3Fio*.
- 2) В папке *p3Fio* создать форму (класс) *fioMenu*.

- 3) На форме *fiоMenu* создать *Метку* с надписью *МЕНЮ fiо*. В *свойствах* изменить: стиль шрифта на *полужирный курсив*; размер – 24; цвет текста (*foreground*) – *красный*.
- 4) Изменить главный класс проекта (запускаемую форму):
 - на имени проекта *ri3Fio* сделать *rClick*, выбрать – *Установить конфигурацию \Настроить*.
 - Указать новый главный класс – возле поля *Главный класс* нажать *Обзор* и выбрать *r3Fio.fiоMenu*, затем, *Выбор главного класса* *Это новый главный класс.
- 5) Подтвердить выбор – *ОК*.
- 6) Запустить приложение на выполнение и проверить загрузку формы *fiоMenu* как главной (стартовой). Остановить приложение.

Дополнительное задание.

- Добавить на форму *fiоMenu* кнопки и события в них для запуска форм *fiоDiagram*, *ing3Fio*. *См. предыдущую лб.
Размер формы *МЕНЮ* уменьшить под кнопки.

После выполнения показать преподавателю.

z4 ИЗ 23 «Меню»

Постановка задачи: создать приложение, включающее меню стартующее при его запуске. Обеспечить запуск из меню форм: “*Диагностика*”; “*База данных*”. Результаты выполнения инд. задания разместить в файле *z4fiоMenu.doc*.

Выполнение.

- 1) Создать проект *z4fiоMenu* и стартовую форму *fiоMENU* в пакете *p4*.
- 2) На форме *fiоMENU* создать:
 - в левом верхнем углу формы разместить компоненту “*метка*” и указать в ней код индивидуального задания – *z4fiоMenu*;
 - в центре компоненту “*метка*” с надписью “*МЕНЮ fiо*”. В *свойствах* изменить: стиль шрифта на *полужирный курсив*; размер – 24; цвет текста (*foreground*) – *красный*;
- 3) Создать дополнительные формы:
 - *fiоDiagnost* и разместить на ней надпись “*Диагностика*”.
 - *fiоDB* и разместить на ней надпись “*База данных*”.
- 4) На форме *fiоMENU* дополнительно создать кнопки:
 - с надписью “*Диагностика fiо*” и событие в ней для запуска формы *fiоDiagnost*;
 - с надписью “*База данных fiо*” и событие в ней для запуска формы *fiоDB*.
- 5) На форме *MENUfiо* создать рамку с надписью “*МенюFio*” и переместить в нее кнопки.
- 6) Запустить приложение, открыть формы “*Диагностика fiо*” и “*База данных fiо*”.
- 7) Сделать копию экрана отформатировать ее и разместить в файле *z4fiоMenu.doc*.
- 8) Сделать копию текста методов по запуску форм из меню и разместить в файле *z4fiоMenu.doc*.
- 9) Файл переслать преподавателю на *dist*.

3. Сервер базы данных

б1 ЛР 31 “Теоретические основы баз данных”

Цель: получить основные представления о базах данных.

Исходные условия: наличие друзей.

Постановка задачи. Разработать формализованную таблицу и заполнить ее данными о друзьях.

Выполнение работы

Теоретические положения

- 1) Реляционная база данных (БД) представлена совокупностью взаимосвязанных таблиц.
- 2) Таблица БД имеет имя и включает коллекцию записей одинаковой структуры, которая представлена перечнем именованных полей.
- 3) Запись включает два вида полей, различающихся по функции:
 - идентификатор – значение поля является уникальным и не может повторяться в других записях;
 - информационные поля – содержат полезные данные. Например, результат Y и время T диагностирования.
- 4) Тип данных указывается в каждом поле и может принимать значение
 - *int* – целочисленное значение;
 - *real* – дробное число
 - *nvarchar 30* текстовое значение размером до 30 символов.
 - др.

Формализация таблицы

- 5) Создание базы данных начинается с разработки таблиц. Пример формального представления таблицы и ее содержания (данные о друзьях) приведены в таблице *FriendFio*.

Таблица - *FriendFio*

Идентификатор	ФИО	Рост, м	Вес, кг
1	Крагельский Ю.А.	1,56	64
2	Шмелев В.А.	2,12	105
3	Фомченкова Р.И.	1,74	59
4	Корнилова У.Н.	1,62	76
<i>fIdFio (int)</i>	<i>fNameFio (varchar 30)</i>	<i>fHfio (real)</i>	<i>dMfio (int)</i>

Дополнительное задание.

Создать таблицу, подобную выше приведенной, с данными о 4х друзьях студента.

После выполнения показать преподавателю.

z5 “Таблица” ИЗ 31 “Формализовать таблицу”

Постановка задачи: на основе графического представления функции сформировать ее табличное представление с кодами таблицы и ее полей. Результаты выполнения инд. задания разместить в файле *z5fioTabl.doc*.

Выполнение.

- 1) Получить у преподавателя графическое представление функции изменения технического состояния детали $Y=f(t)$.
- 2) На функции равномерно разместить 9ть точек. *Всего получится 11 точек включая начало и конец функции.
- 3) Формализованные данные точки включают: идентификатор; значения Y,t ; примечания.
- 4) Нарисовать таблицу *DiagnostFIO* с полями для размещения данных о точках. В имена ее полей добавить *fio*.
- 5) Значения точек на границах интервалов занести в *DiagnostFIO* – будет сформировано 11 строчек. В любых двух строчках указать примечания (например, высокая температура, критическая вибрация, скрежет, удары). *Точки начала и конца функции указать обязательно.
- 6) Сделать фотографию графического и табличного представления функции и вставить в файл *z5fioTabl.doc*.
- 7) Указать код, наименование работы и исполнителя.
- 8) Файл переслать преподавателю на *dist*.

62 ЛР 32 “Запуск СУБД Java DB” db sample

Цель: используя средства *NetBeans* подключить и запустить систему управления базой данных (СУБД) *JavaDB*.

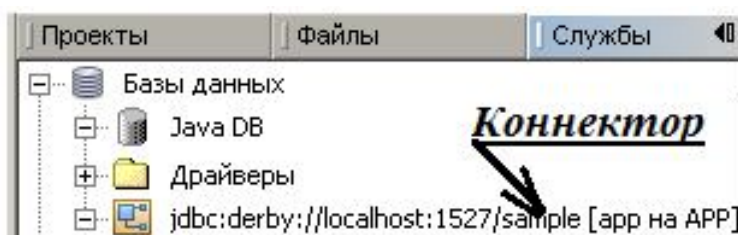
Исходные условия: наличие *NetBeans* с установленной базой данных *JavaDB*.

Постановка задачи. Выполнить доступ к информации учебной базы данных *sample*.

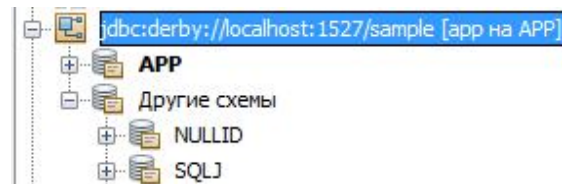
Выполнение работы

Подключение к базе данных (БД) *sample* на основе СУБД *JavaDB*

- 1) Запустить *NetBeans*.
- 2) На левой панели открыть вкладку *Службы*.
- 3) Найти на ней папку *Базы данных / javaDb / sample*.
- 4) На *sample* сделать *rClick* и выбрать (>) *Установить соединение*. *Появится коннектор.
- 5) По *rClick* на коннекторе *sample* можно установить или разорвать соединение с базой данных *sample*.

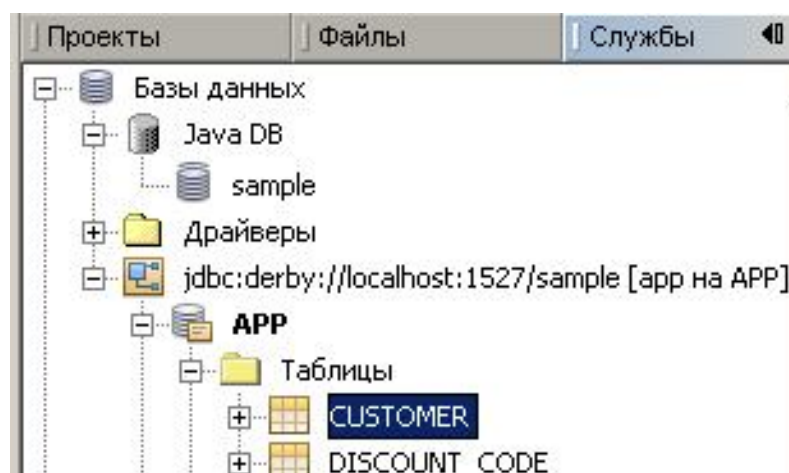


- 6) Соединиться с БД *sample*. Сделать на коннекторе *rClick* > *Установить соединение*.
 *Будет устранен разрыв на пиктограмме коннектора
- 7) Открыть компоненты базы данных *sample*, (нажать + в строке коннектора) см.рисунок.

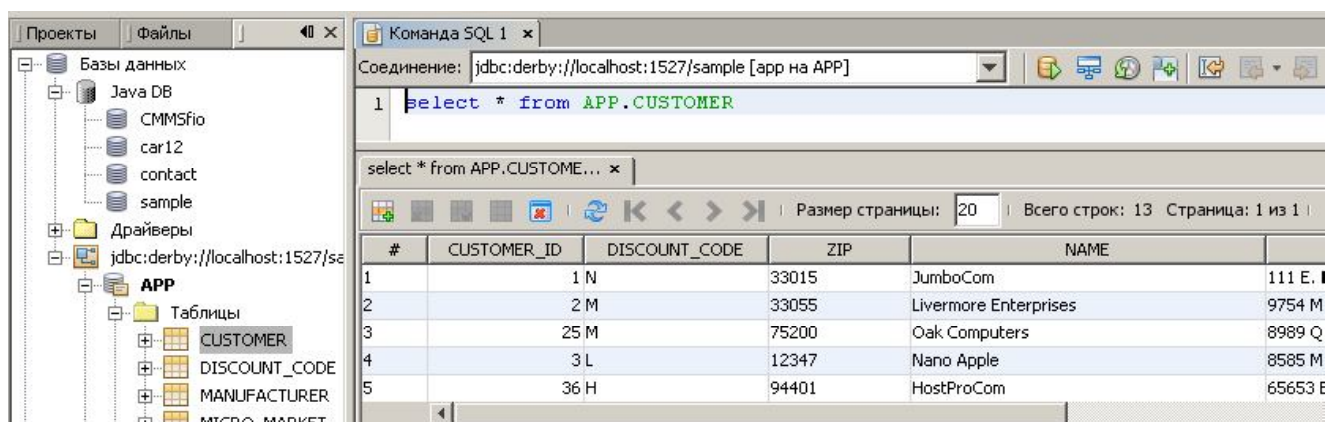


Доступ к содержанию БД

- 8) Открыть перечень таблиц БД *sample* (нажав + на *APP / Таблицы*), см.рисунок .



- 9) Открыть содержание таблицы *CUSTOMER*
- Сделать *rClick* на наименовании таблицы *CUSTOMER* > *Просмотр данных*.
 - *На панели редактора будет создана вкладка и на ней откроется запрос к БД на чтение таблицы *select * from APP.CUSTOMER* и все записи таблицы *CUSTOMER*. см.рисунок.



- 10) Работа с соединением с БД.

- Разорвать соединение - указать коннектор *sample* (пиктограмма со связью), сделать *rClick > Разорвать соединение* . *В пиктограмме оборвутся связи.
- Соединиться с БД - указать коннектор *sample* (пиктограмма со связью), сделать *rClick > Установить соединение* . *В пиктограмме соединятся связи и будет открыт доступ к ее таблицам.

*Примечания. Файлы баз данных располагаемые в папке

☒ (XP) *сист.диск:\Documents and Settings\admin\.netbeans-derby*.

☒ (Seven) *сист.диск:\Users\home\.netbeans-derby*.

☒ *В этой папке расположена и БД *sample* представленная папкой *sample*.

Дополнительное задание.

- В БД *sample* в таблице *MANUFACTURER* найти запись с идентификатором равным номеру студента в журнале преподавателя и записать в конспект телефон производителя (поле *Phone*)

После выполнения показать преподавателю.

63 ЛР 33 “Создание БД” db DBfio

Цель: используя средства *NetBeans* создать новую БД под управлением *JavaDB*.

Исходные условия: наличие *NetBeans* с установленной *JavaDB*.

Постановка задачи. Создать БД «*DBfio*» с таблицей «*FriendFio*».

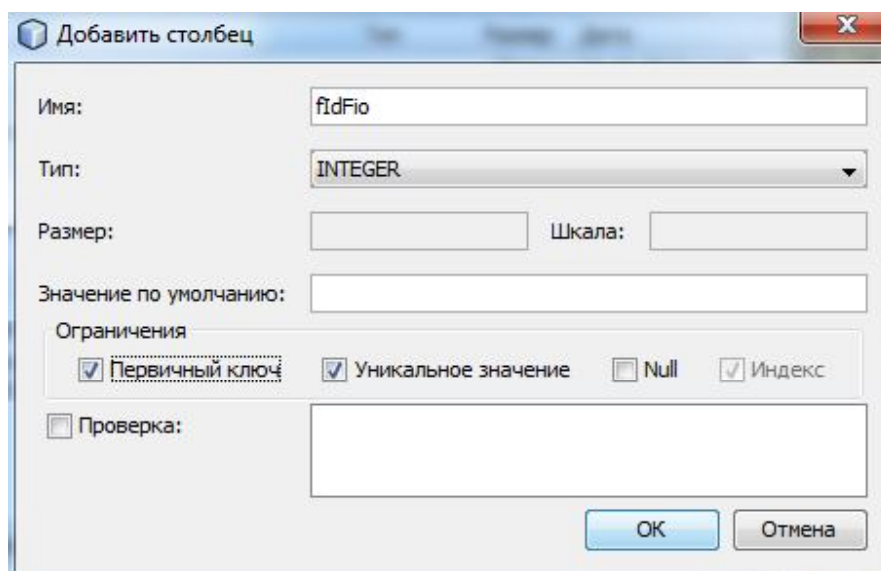
Выполнение работы

Создание базы данных

- 1) Начать создание - На панели *Службы* выбрать *JavaDB* сделать *rClick > Создать базу данных*.
 - 2) На панели *Создать базу данных Java DB* указать параметры:
 - Имя базы данных = *DBfio*
 - Имя пользователя – не заполнять;
 - Пароль - не заполнять
 - Нажать кнопку *ОК*.
- *В перечне баз данных (в ветви *Java DB*) откроется новая запись с именем созданной БД и будет создан коннектор к ней.
- 3) Соединить БД *DBfio* с *NetBeans* - *rClick* на ее коннекторе *>Установить соединение >ОК*. *Изменится пиктограмма коннектора.

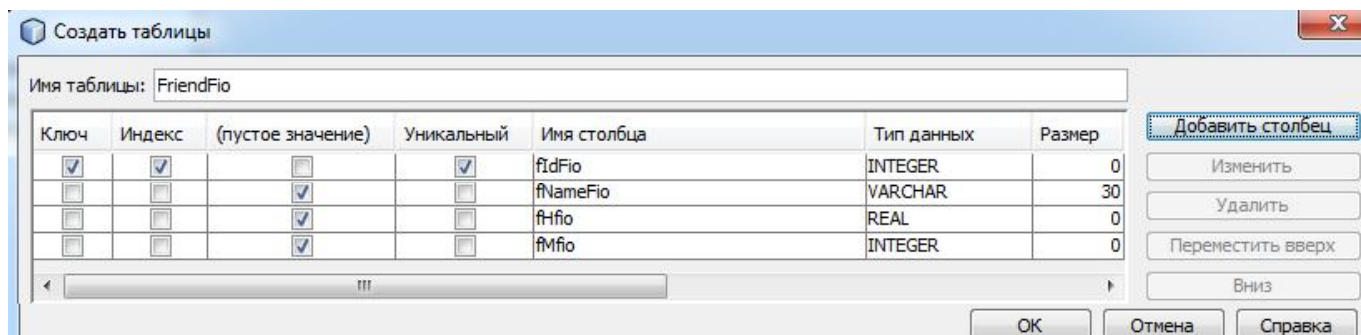
Создание таблицы в базе данных.

- 4) Указать папку таблиц – *раскрыть* коннектор *DBfio* затем *APP> Таблицы*.
- 5) Начать создание таблицы – *rClick* на *Таблицы > Создать таблицу*. *Откроется панель для создания новой таблицы.
- 6) Присвоить имя таблице – на панели *Создать таблицы* указать параметр Имя таблицы = *FriendFio*.
- 7) Создать столбец идентификатора – нажать кнопку *Добавить столбец* и присвоить параметры: Имя = *fldFio*; Тип = *INTEGER*; Первичный ключ – *выбрать*.



- 8) Создать столбец – нажать *OK*.
- 9) Создать столбец *fNameFio* типа *VARCHAR*, Размер – 30.
- 10) Подтвердить создание таблицы – на панели *Создать таблицы* нажать кнопку *OK*
- 11) Создать столбец *fHfio* типа *REAL*.
- 12) Подтвердить создание таблицы – на панели *Создать таблицы* нажать кнопку *OK*
- 13) Создать столбец типа *INTEGER*.

*В результате будет получен перечень столбцов и их характеристики, см. рисунок



- 14) Подтвердить создание таблицы – на панели *Создать таблицы* нажать кнопку *OK*. *На панели *Службы*, в ветви коннектора *DBFio* появится таблица *APP>Таблицы>FriendFio* (нажать +) с созданными полями.
- 15) В коннекторе *DBFio* разорвать соединение.
- 16) В коннекторе *DBFio* установить соединение, открыть перечень полей.
- 17) Открыть характеристику поля *fIdFio* - *rClick* на нем, выбрать *Свойства*. Ознакомиться и закрыть.

Дополнительное задание.

- В таблицу *FriendFio* добавить поле *checkFio* типа *VARCHAR*, Размер – 17.

После выполнения показать преподавателю.

64 ЛР 34 “Создание записей в таблице” db «DBfio»

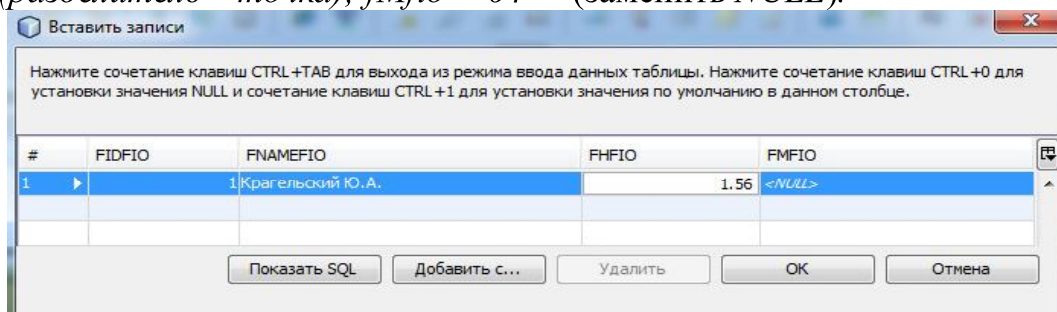
Цель: используя средства *NetBeans* создать записи в таблице БД.

Исходные условия: наличие *NetBeans* с установленной СУБД *JavaDB*, и БД «*DBfio*» с таблицей «*FriendFio*».

Постановка задачи (ПЗ). В таблицу «*FriendFio*» ввести данные о друзьях из б1 (ЛР 31).

Выполнение работы

1. Загрузить БД *DBfio* и открыть содержание таблицы *FriendFio*. *Она без записей.
2. Открыть вставку новых записей – нажать зеленый «+» слева над таблицей. *Откроется панель «Вставить записи».
3. Сформировать 1ю запись - взять значения из первой строки таблицы созданной в б1 (ЛР 31). Присвоить полям: *fldFio* = 1; *fName*= Крагелский Ю.А. ; *fHfio* = 1.56(разделитель – точка); *fMfio* = 64 - (заменить *NULL*).



4. Создать бланк 2й записи – под таблицей нажать кнопку (Добавить с...).
5. Ввести во вторую запись значения из второй строки таблицы в ЛР 21.
6. Ввести созданные записи в базу данных – нажать «Ок». *Введенные две записи будут видны в таблице

#	FIDFIO	FNAMEFIO	FHFIO	FMFIO
1		1 Крагелский Ю.А.	1.56	64
2		2 Шмелев В.А.	2.12	105

7. Открыть вставку новых записей.
8. Аналогично ввести третью запись из таблицы в ЛР 31. *При вводе обеспечить уникальность значения поля *fldFio* – идентификатора записи в таблице *FriendFio*.
9. В коннекторе *DBFio* разорвать и установить соединение. Ввести прочие записи из таблицы в ЛР 31 и сохранить их.
10. В записи *Шмелев В.А.* корректировать его рост. В поле *FHFIO* на значении 2.12 сделать *wClick* и ввести значение 1.95 (разделитель – точка). Для сохранения в поле – *Enter*. Для сохранения в таблице – в меню кнопка *Фиксировать записи*.

Дополнительное задание.

- В таблицу *FriendFio* ввести запись с данными студента выполнявшего работу..

После выполнения показать преподавателю.

z6 ИЗ 32 “Создание БД”

Постановка задачи: Создать базу данных и разместить в ней данные диагностирования из таблицы, разработанной в *z5fioTabl*. Результаты выполнения инд. задания разместить в файле *z6fioDB.doc*.

Выполнение.

- 1) Создать БД *z6fioDiagn*.
- 2) В БД *z6fio* создать таблицу. Ее имя, имена и характеристики ее полей взять из таблицы в индивидуальном задании *z5fioTabl*.
- 3) Сформировать записи в БД *z6fioDiagn > DiagnostFIO* на основе записей таблицы в индивидуальном задании *z5fioTabl*.
- 4) Открыть вкладку *Службы и Записи таблицы*. Сделать копию экрана и **разместить** ее в файле *z6fioDB.doc*.
- 5) Файл переслать преподавателю на *dist*.

4. Приложение для работы с БД

п1 ЛР 41 “Создать приложение «Перечень друзей» ” rP1FriendFio

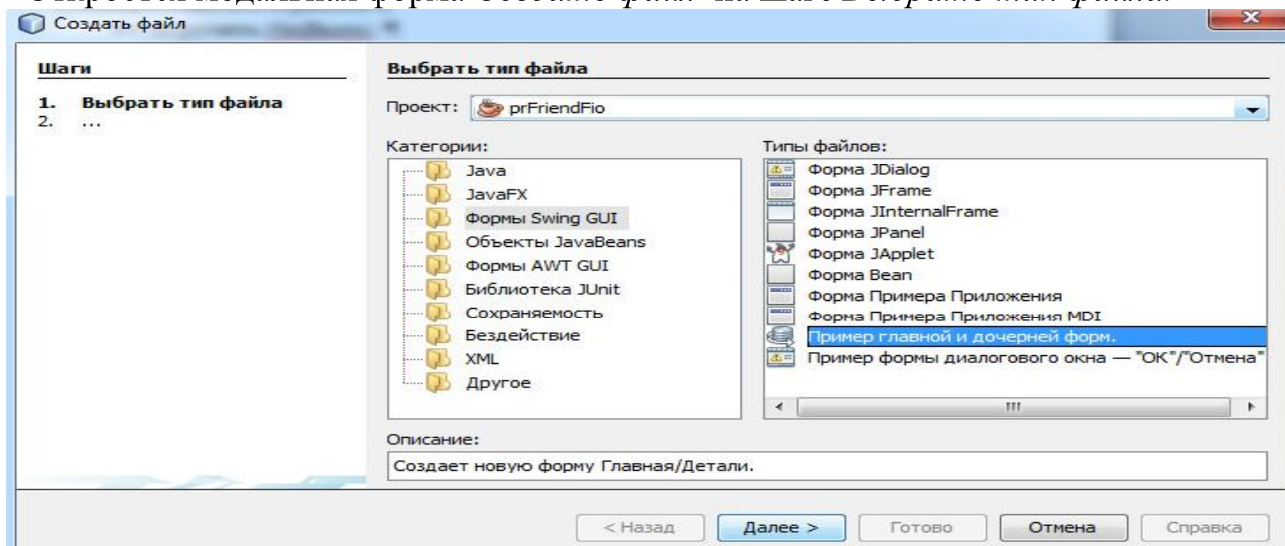
Цель: получить навыки создания простейшего приложения, работающего с базой данных.

Исходные условия: наличие базы данных *DBfio* с таблицей *FriendFio*.

Постановка задачи (ПЗ). Создать приложение для ввода и редактирования данных о друзьях.

Выполнение работы

- Подготовиться к работе:
 - загрузить студию *NetBeans*;
 - установить соединение с *DBfio* – выполнить на панели *Службы*;
 - открыть вкладку *Проекты* – размещена на левой панели.
 - Выбрать вид проекта:
 - Начать создание *Файл > Создание проекта*. Главный класс не создавать;
 - Выбрать вид проекта - *Java* (в Категории) и *Приложение Java* (в Проекты), Нажать *Далее*.
 - Создать новый проект с именем *rP1FriendFio*.
 - Создать форму (класс) с таблицей:
 - начать создание – *rClick* на имени проекта *rP1FriendFio > Новый > Другое*.
- *Откроется модальная форма *Создать файл* на шаге *Выбрать тип файла*.



На форме указать:

- Категория - *Формы Swing GUI*;
 - Типы файлов - *Пример главной и дочерней форм*;
 - Нажать *Далее*.
- * Откроется модальная форма *Создать Главная/дочерняя формы* на шаге *Имя и расположение*.

На форме указать:

- Имя класса – *fFriendFio*;
- Пакет – *pFriendFio*;
- Нажать *Далее*.

*Откроется шаг *Главная таблица*

На форме указать:

- Подключение к базе данных - (выбрать конектор ... *DBFio...*;
- Таблица базы данных - выбрать таблицу *FRIENDFIO*. *Откроется перечень столбцов таблицы выбранных для вывода на форму;
- Нажать *Далее* > *Готово*.

*Автоматически будет сгенерирован класс (форма) с таблицей .

5. Запустить приложение на выполнение. *На экране появится форма, показанная на рисунке.

Fidfio	Fnamefio	Fhfio	Fm
1	Крагельский...	1,56	64
2	Шмелев В.А.	2,12	105
3	Фомченкова ...	1,74	59
4	Корнилова У...	1,62	76

Fidfio:

Fnamefio:

Fhfio:

Fm:

New Delete Refresh Save

6. Закрыть приложение

После выполнения показать преподавателю.

п2 ЛР 42 “Работа в приложение «Перечень друзей» ” rP1FriendFio

Цель: получить навыки использования приложения, работающего с базой данных.

Исходные условия: наличие базы данных *DBfio* с таблицей *FriendFio*. и приложения *rP1FriendFio*.

Постановка задачи (ПЗ). В приложении выполнить редактирование полей таблицы и создание новых записей.

Выполнение работы

1. Подготовиться к работе - загрузить и запустить приложение *rP1FriendFio*.
2. Выполнить редактирование данных:
 - в таблице установить курсор во вторую сверху строку. *В нижних полях появятся значения ее полей;
 - выполнить редактирование - в поле *fMfio* ввести «115»;
 - сохранить редактирование – нажать кнопку *Save*;
 - обновить таблицу данными с сервера - нажать кнопку *Rrefresh* (изменений не будет).
3. Создать новую запись (нажать *New*), заполнить значениями, сохранить и обновить.
*Обратить внимание – значение в поле *fidFio* должно быть уникальным
4. Не останавливая приложения *rP1FriendFio* выполнить редактирование данных непосредственно в базе данных (см. б4 ЛР 34 “Создание записей в таблице»):
 - перейти в студию базы данных *Службы* и подключить базу данных *DBfio* - wClick на ее коннекторе.
 - Нажимая «+» найти таблицу *FriendFio*.
 - На таблице rClick «Просмотр данных» - загрузить записи таблицы.
 - Создать дополнительную запись с данными преподавателя.
 - Сохранить запись и разорвать соединение в коннекторе.

5. Вернуться в запущенное приложение и обновить таблицу данными с сервера - нажать кнопку *Rrefresh* (станет видна добавленная запись).
6. Остановить приложение.
7. Ознакомиться со структурой проекта.
 - Закрыть все вкладки в редакторе классов.
 - Найти класс *FriendFio* (коды для подключения к базе данных) - /Пакеты исходных классов/ *pFriendFio / FrendFio.java* и загрузить в редактор. *Посмотреть операторы, обеспечивающие подключение компонент на форме к полям в таблице базы данных. *Найти строку обеспечивающую чтение всех записей из таблицы в базе данных – код `@NamedQuery(name = "Friendfio.findAll", query = "SELECT f FROM Friendfio f")`
 - На форме *fFriendFio* найти и ознакомиться с событиями в кнопках New, Save, Refresh.

Дополнительное задание.

- Добавить данные о двух друзьях. *Всего должно быть 9 записей.
- Переместить компоненты на форме, изменить надписи на кнопках. *Должен получиться интерфейс, показанный на рисунке.



- Запустить приложение и проверить работу.

После выполнения показать преподавателю.

z7 «Диагностика» ИЗ 41 “Создать приложение «Диагностика»”

Постановка задачи: Создать приложение для работы с записями таблицы *DiagnostFIO* из базы данных *zbfio*, разработанной в *zbfio*. Результаты выполнения инд. задания разместить в файле *z7fioDiagnost.doc*.

Выполнение.

1. Создать приложение *z7fioD* для работы с таблицей *DiagnostFIO* из БД *z32fio*, созданной при выполнении *zbfio*.
2. Интерфейс формы для вывода данных таблицы выполнить подобным принятому в п2 (ЛР42).
3. Запустить приложение, сделать копию формы, отформатировать ее **разместить** в файле *z7fioDiagnost.doc*.

Файл переслать преподавателю на *dist*.

п3 ЛР 43 “Чтение поля из таблицы базы данных” rP3fioRead

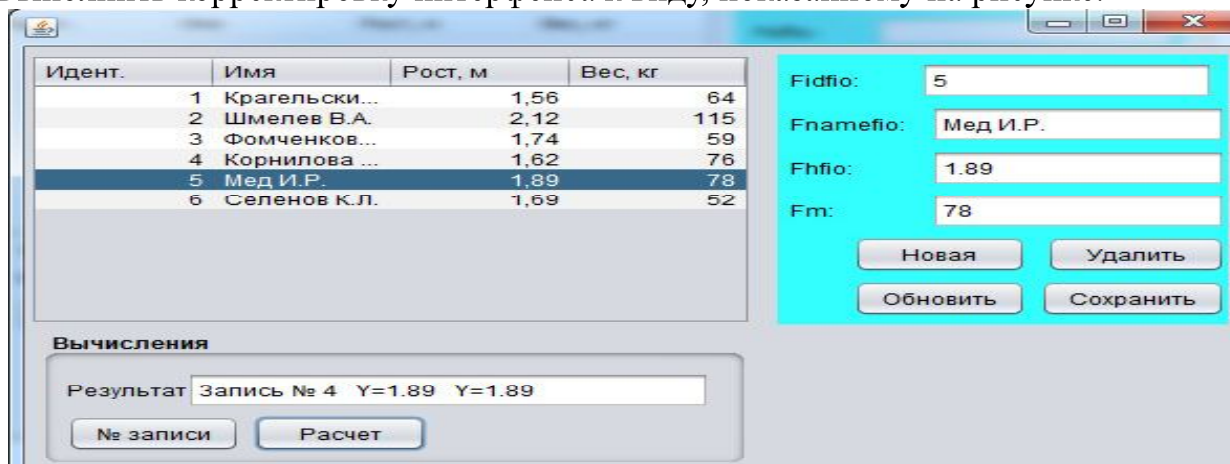
Цель: освоить средства для чтения значения поля из записи в таблице базы данных.

Исходные условия: наличие базы данных *DBfio* с данными о друзьях в таблице *FriendFio*.

Постановка задачи. Создать приложение читающее из текущей записи значение поля *fHfio*, возводящее его в квадрат и выводящее результат на экран.

Выполнение работы

- Подготовиться к работе:
 - на панели *Службы* установить соединение с базой данных *DBfio*.
- Создать новое приложение:
 - Начать создание *Файл > Создать проект*.
 - выбрать *Java* (в Категории) и *Приложение Java* (в Проекты). Нажать *Далее*.
- Указать: имя проекта *rP3fioRead*; не выбирать «Использовать отдельную папку для библиотек» и «Создать главный класс». Нажать *Готово*.
- Создать форму (класс) с таблицей:
 - начать создание – *rClick* на имени проекта *rP3fioRead > Новый > Другое*.
 - Выбрать: Категории = *Формы Swing GUI*; Типы файлов = *Пример основной/подробной формы*. Нажать *Далее*.
 - Ввести. Имя класса = *fP3FriendFio*; Пакет = *rP3fio*. Нажать *Далее*.
 - Подключить таблицу: активировать коннектор ... *DBFio...*; выбрать таблицу *FRIENDFIO*. Нажать *Далее*, затем *Готово*.
- Для проверки запустить приложение на выполнение. *На экране появится форма, показанная на рисунке.
- Выполнить корректировку интерфейса к виду, показанному на рисунке.



Для этого.

- Справа от таблицы создать карточку (из панели) на нее перенести компоненты из-под таблицы.
- Под таблицей создать панель с надписью «п3 Вычисления» и в ней разместить компоненты:
 - метку *otchet* текстом *Результат*;
 - текстовое поле *RezFio* с текстом ????????

- кнопку *nomZapFio* с надписью № записи и Свойства / *toolTipText* = Номер текущей записи в таблице;
 - кнопку *RaschetFio* с надписью Расчет и Свойства / *toolTipText* = Дописать рост.
9. В кнопке *nomZapFio* создать событие *MouseClicked*. В событии прописать код обеспечивающий вывод номера текущей записи в таблице, указанной в окне данных (таблице) на экране. Должно получиться подобное нижеприведенному.

```
{
    int nZap;
    nZap = masterTable.getSelectedRow();//номер текущей записи
    RezFio.setText("Запись № " + String.valueOf(nZap));//вывод на форму
}
```

10. В кнопке *RaschetFio* создать событие *MouseClicked*. В событии прописать код обеспечивающий вывод значение роста из текущей записи в таблице, указанной в окне данных на экране. Должно получиться нижеприведенное.

```
{
    //чтение из таблицы БД _____
    int nZap; //переменная для номера записи
    nZap = masterTable.getSelectedRow();//номер текущей записи
    Friendfio d;//создание бланка записи из полей таблицы Friendfio
    //запись в бланк данных текущей строки
    d = list.get(masterTable.convertRowIndexToModel(nZap));

    //чтение данных с компоненты формы _____
    String mark; //создание строковой переменной
    mark = RezFio.getText();//присвоение значения

    //обработка данных (инж.задача) _____
    float y;
    y = d.getFnfio();//значение поля fnfio из текущей записи таблицы
    mark = mark + " Y=" + String.valueOf(y);//корректировка переменной

    //вывод результата на форму (на экран) _____
    RezFio.setText(mark); //присвоение значения компоненте RezChen на форме }
}
```

Дополнительное задание.

- Создать кнопку с надписью *ДопЗадание* и событие в ней для вывода в *RezFio* фамилии и частного от деления массы на рост.
- Запустить приложение и проверить работу.

После выполнения показать преподавателю.

п4 ЛР 44 “Знакомство с циклом FOR” rP3fioRead

Цель: освоить совместный анализ данных из нескольких строк в таблице базы данных.

Исходные условия: наличие базы данных *DBfio* с данными о друзьях и проекта *rP3fioRead* с формой из предыдущей лабораторной работы.

Постановка задачи (ПЗ). Создать приложение: читающее значение поля *fHfio* из заданных записей таблицы *FRIENDFIO*, определяющее сумму значений *fHfio* и выводящее результат на экран.

Выполнение работы

1) Теория *for* ($i = 0; i < k; i++$) { операторы }:

- i – номер шага в цикле;
- k – максимальное возможное число шагов;
- $i++$ - увеличение шага.

2) Подготовиться к работе:

- на панели *Службы* установить соединение с базой данных *DBfio*,
- загрузить проект *prP3fioRead*.

3) Освоить выделение некоторых записей в окне данных. Запустить приложение. В окне данных на необходимых записях нажать *Shift+Click*, *Ctrl+Click*. * Выделенные записи станут темными.

4) Остановить проект.

5) На форме *fP3FriendFio* создать:

- панель с надписью *n4 Выбранные записи*;
- на панели кнопку *sumHfio* с надписью *Сумма H*;
- на панели под кнопкой текстовое поле *rezHfio*.

6) В кнопке *sumHfio* создать событие *MouseClicked*. В событии прописать код обеспечивающий суммирование выбранных записей таблицы *masterTable*. Должно получиться подобное нижеприведенному.

```
{
    //внутренняя переменная
    float sumFio = 0;
    int kolFio;
    //формирование перечня номеров выделенных строк _____
    int[] selectedStr;//создание массива для номеров выделенных строк
    selectedStr = masterTable.getSelectedRows();//заполнение массива selectedStr
    kolFio = selectedStr.length; //количество выбранных записей
    //последовательный анализ выбранных строк из таблицы _____
    for (int n= 0; n < kolFio; n++) {
        //чтение в бланк d полей строки с номером selectedStr[idx]
        Friendfio d;//создание бланка
        d = list.get(masterTable.convertRowIndexToModel(selectedStr[n])); //заполнение
        бланка данными n-й выделенной записи таблицы
        sumFio = d.getFhfio() + sumFio; //добавление значения поля в сумму
    }
    //вывод результата на экран после выхода из цикла for _____
    rezHfio.setText("СуммаH = " + String.valueOf(sumFio)); //вывод суммы на форму
}
```


Дополнительное задание.

- Модифицировать событие *MouseClicked* для определения и вывода на экран среднего значения из выбранных строк.
- Запустить приложение и проверить работу.

После выполнения показать преподавателю.

п5 ЛР 45 “Знакомство с условием IF” rP3fioRead

Цель: освоить совместный анализ всех строк таблицы базы данных и запись результата анализа в строки таблицы.

Исходные условия: наличие базы данных *DBfio* с данными о друзьях и проекта *rP3fioRead* с формой из предыдущей лабораторной работы.

Постановка задачи (ПЗ). Создать приложение: читающее значение поля *Fhfio* из всех записей, определяющее среднее значение для них; выводящее в поле *Fnamefio* записей таблицы значение "м" – если *Fhfio* меньше среднего и значение "б" – если больше.

Выполнение работы

1) **Теория** *if (i < k) { операторы 1} else { операторы 2} :*

- если $(i < k)$ - выполняется скобка { операторы 1};
- иначе - выполняется скобка { операторы 2}.

2) Подготовиться к работе:

- на панели *Службы* установить соединение с базой данных *DBfio*,
- загрузить проект *rP3fioRead*;
- загрузить в редактор класс *fP3FriendFio.java*.

3) Создать на форме:

- панель с надписью *Все записи*;
- на панели кнопку *allZapFio* с надписью *n5 ВсеЗаписи*.
- на панели под кнопкой текстовое поле *allHfio*.

4) В кнопке *allZapFio* создать событие *MouseClicked*. Обеспечивающий

- суммирование значений поля *Fhfio* в текущей и всех предыдущих записях таблицы;
- вывод накопленной суммы в поле *checkFio* каждой записи;
- вывод суммы всех записей в поле *allHfio*.

Должно получиться подобное нижеприведенному

```
{
    int kolRowFio = 0;
    float sumFio = 0;
    kolRowFio = masterTable.getRowCount();// количество записей в таблице masterTable
    for (int idx = 0; idx < kolRowFio; idx++) {
        //чтение строки из таблицы
```

```

    Friendfio d = list.get(masterTable.convertRowIndexToModel(idx));

    sumFio = sumFio + d.getFhfio();//суммирование
    d.setCheckfio( String.valueOf(sumFio)); //запись суммы в таблицу
}
allHfio.setText ("Сумма Н= " + String.valueOf(sumFio)); //вывод на форму
}
5) Модифицировать MouseClicked. В событии прописать код обеспечивающий:


- определение среднего значения Fhfio по всем записям;
- сравнение в каждой записи Fhfio со средним и дописывающее в поле CheckFio записи символ "м__" – при Fhfio < среднего и "б. ." в противном случае.
- Должно получиться подобное нижеприведенному.



```

{
 int kolRowFio = 0;
 float sumFio = 0;
 kolRowFio = masterTable.getRowCount();// количество записей в таблице masterTable
 for (int idx = 0; idx < kolRowFio; idx++) {
 //чтение строки из таблицы
 Friendfio d = list.get(masterTable.convertRowIndexToModel(idx));

 sumFio = sumFio + d.getFhfio();//суммирование
 d.setCheckfio(String.valueOf(sumFio)); //запись суммы в таблицу
 }
 allHfio.setText ("Сумма Н= " + String.valueOf(sumFio)); //вывод на форму
 //*****модификация
 float srednFio = sumFio/kolRowFio;
 for (int idx = 0; idx < kolRowFio; idx++){
 //чтение строки из таблицы
 Friendfio d = list.get(masterTable.convertRowIndexToModel(idx));
 //анализ
 if (d.getFhfio() < srednFio)
 d.setCheckfio(" м__");
 else d.setCheckfio("б. .");
 }
 allHfio.setText ("Средн Н= " + String.valueOf(srednFio)); //вывод на форму
 //!!!Запрещается нажимать кнопку Save }

```


```

Дополнительное задание.

- Дополнительно обеспечить вывод в поле *checkFio* таблицы разницу между средним значением и *Fhfio*.
- Запустить приложение и проверить работу.

После выполнения показать преподавателю.

z8 “Средняя V” ИЗ 42 “Средняя скорость изменения Y”

Постановка задачи: разработать приложение для определения средней скорости изменения технического состояния детали, как среднего значения скоростей между результатами измерений на основе данных таблицы *fioDiagnostFIO* из базы данных *zbfio*. Результаты выполнения инд. задания разместить в файле *z8fioSrednV.doc*.

Выполнение.

- 1) Создать приложение *z8fioV* для работы с таблицей *DiagnostFIO* из БД *zbfio*, созданной при выполнении *zbfio*.
- 2) Интерфейс формы *f8fioV* для вывода данных таблицы выполнить подобным принятому в *z7fioD*.
- 3) На форме *f8fioV* разместить панель с надписью « *z8fioV* “Средняя скорость изменения Y” ».
- 4) На панели разместить:
 - кнопку для запуска метода по определению среднего значения V;
 - текстовое поле для вывода результата.
- 5) Разработать метод для решения задачи.
- 6) Запустить приложение и результат вывести на экран.
- 7) Сделать копию запущенного приложения и **разместить** ее в файле *z8fioSrednV.doc*.
- 8) Сделать копию формы, отформатировать ее **разместить** в файле *z8fioSrednV.doc*.
- 9) В файл *z8fioSrednV.doc* **добавить** копию метода по определению среднего значения.
- 10) Файл переслать преподавателю на *dist*.

5. Численный метод решения задачи

ч1 ЛР 51 “Использование численного метода”

Цель: ознакомиться с математическим аппаратом численного метода, обеспечивающего решение задачи путем последовательного приближения.

Исходные условия: знание алгебраического метода решения системы двух линейных уравнений.

Постановка задачи. Решить систему 2х уравнений методом последовательного приближения.

Выполнение работы

Алгебраический метод решения системы

1) Система линейных уравнений:

$$y = f1(t) = 3 + 2 * t$$

$$y = f2(t) = 15 - 3 * t$$

2) Решение системы с использованием алгебраического метода – найти t при $y = y$:

- $3 + 2 * t = 15 - 3 * t$;
- $15 - 3 = 2t + 3t$;
- $12 = 5t$;
- $T = 12/5 = 2,4$.

3) Система нелинейных уравнений:

$$y = 3,6^t + 2 * t$$

$$y = \ln(t) - 3/(t * t * t)$$

Алгебраический метод сложен или не применим.

Графический метод решения первой системы

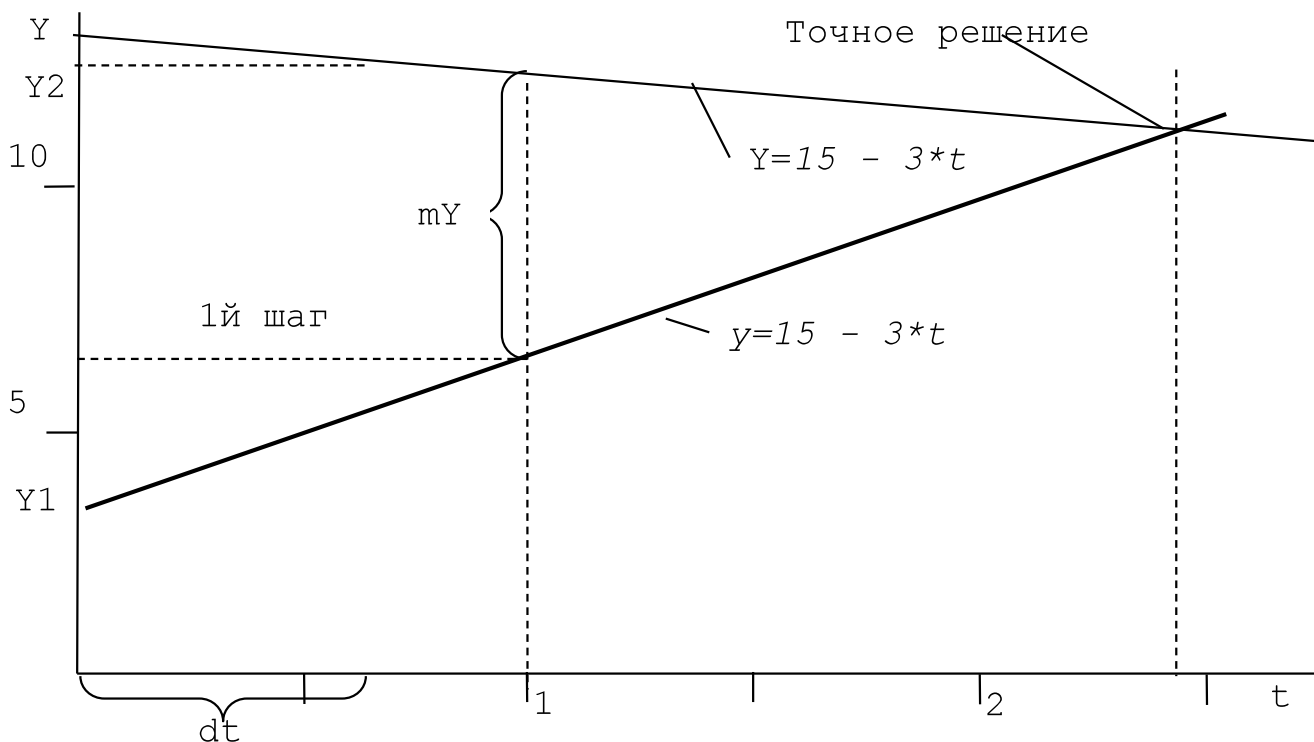


Рисунок 1. Графическое решение системы 2х уравнений.

Теория численного метода решения системы

4) Для решения системы 2х уравнений:

$$y=f1(t)$$

$$y=f2(t),$$

где при начальном значении аргумента $t=t0=0$ определяется начальная ошибка $my0$ из сравнения функций $my0 = f1(t0) - f2(t0)$.

5) Метод последовательного приближения к решению выполняется:

- с шагом dt по аргументу;
- на каждом N -м шаге определяется новое значение аргумента $tN = t(N-1) + dt$, где $t(N-1)$ значение аргумента на предыдущем шаге.

6) Критерий получения решения системы (задания):

- на N -м шаге определить ошибку (разницу) myN между значениями функций системы $myN = f1(tN) - f2(tN)$;
- оценить изменение знака ошибки $zmN = my0 * myN$;
- принять решение - при $zmN < 0$ – решение найдено.

7) Применимость численного метода – применим для любой системы уравнений.

Использованием численного метода для решения системы линейных уравнений

$$y=f1(t) = 3 + 2*t$$

$$y=f2(t) = 15 - 3*t$$

8) Начальные значения- нулевой шаг $N=0$:

- шаг по аргументу $dt = 1$,
- начальное значение аргумента $t0=0$;
- начальная ошибка $my0=f1(t0)-f2(t0)=(3+2*t0)-(15-3*t0)=3+2*0-15+3*0 = -12$.

9) Шаг первый $N=1$ численного решения:

- значение аргумента $t1 = t0 + dt = 0+1=1$;
- ошибка решения $my1=f1(t1)-f2(t1)=(3+2*t1) - (15 - 3*t1) = 3+2*1-15+3*1 = -7$;
- изменение знака $zm1 = my0 * my1 = (-12) * (-7) = 84$;
- вывод $zm1 > 0$ – решение не найдено, делать следующий шаг.

10) Шаг второй $N=2$ численного решения:

- значение аргумента $t2 = t1 + dt = 1+1=2$;
- ошибка решения $my2=f1(t2)-f2(t2)=(3+2*t2) - (15 - 3*t2) = 3+2*2-15+3*2 = -2$;
- изменение знака $zm2 = my0 * my1 = (-12) * (-2) = 24$;
- вывод $zm2 > 0$ – решение не найдено, делать следующий шаг.

11) Шаг третий $N=3$ численного решения:

- значение аргумента $t3 = t2 + dt = 2+1=3$;
- ошибка решения $my3=f1(t3)-f2(t3)=(3+2*t3)-(15 - 3*t3) = 3+2*3-15+3*3 = \underline{+3}$;
- изменение знака $zm3 = my0 * my3 = (-12) * (+3) = -36$;
- вывод $zm = -36 < 0$ – решение найдено.

12) Результат решения:

- значение аргумента $t = 3$;
- ошибка решения $my = +3$;
- количество шагов $N=3$.

13) Время отказа $t = 3$ - результат решения системы уравнений.

14) Табличное представление результатов численного метода.

$dt=1; t_0=0;$

№ шага	t	my	zm	Вывод
0	0	-12		Сл. шаг
1				
2				
3	3	+3	-36	Дата отказа

Дополнительное задание.

- В таблицу ввести результаты 1 и 2 шагов

После выполнения показать преподавателю.

ч2 ЛР 52 “Поиск экстремума”

Цель: освоить использование численного метода для поиска экстремума.

Исходные условия: знание теории численного метода.

Постановка задачи (ПЗ). Найти численным методом максимального значения (экстремума) полинома 2й степени $y=a_1 + b_1*t - c_1*t*t$, где $a_1=0,2; b_1=4; c_1=1,1$.

Выполнение работы

Графический метод решения задачи поиска экстремума.

1) Выражение полинома $y=a_1 + b_1*t - c_1*t*t$, где $a_1=0,2; b_1=4,2; c_1=1,1$.

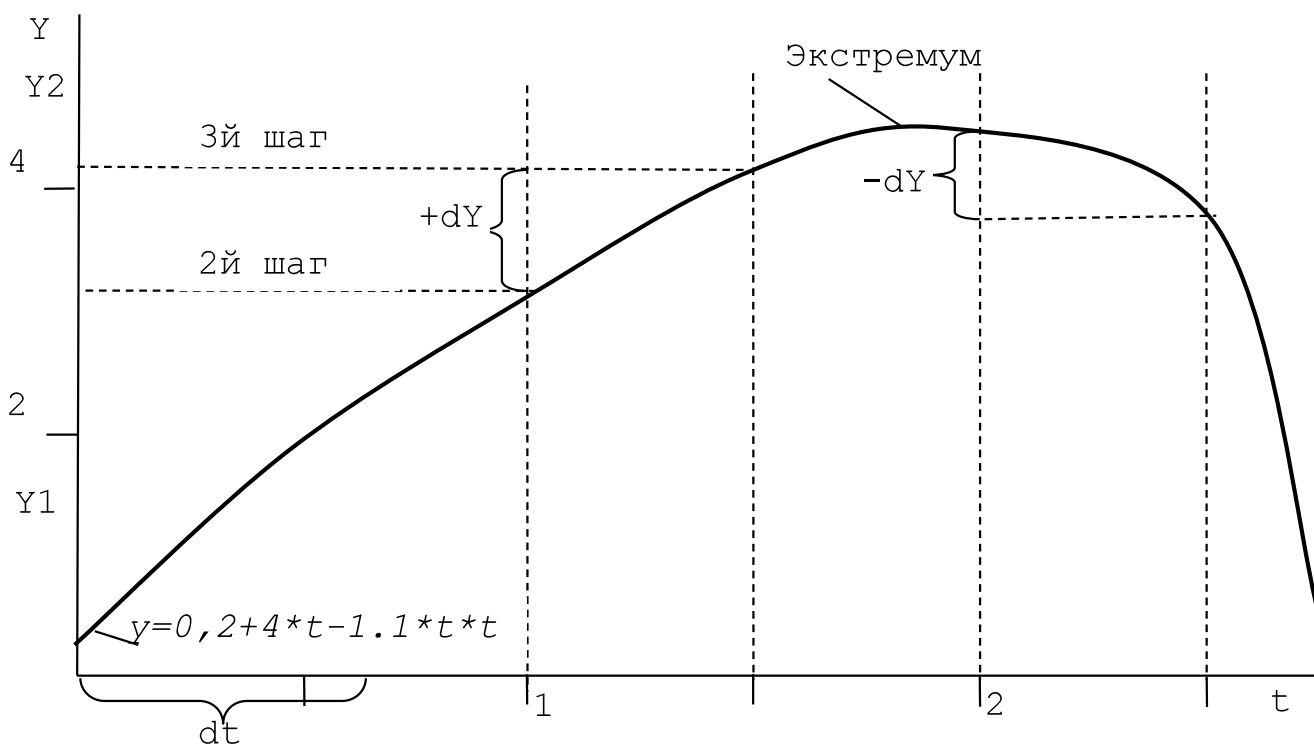


Рисунок 1. Графическое решение поиска экстремума.

Математический аппарат численного метода решения задачи поиска экстремума.

- 2) Выражение полинома $y = a1 + b1*t - c1*t*t$
 - где $a1 = 0,2$;
 - $b1 = 4$;
 - $c1 = 1,1$.
- 3) Исходные параметры аргумента:
 - начальное значение аргумента $t = 0$;
 - шаг по аргументу, $dt = 0,5$.
- 4) Критерий решения задания:
 - определить знак приращения функции на шаге $dy = yN - y(N-1)$;
 - где N номер шага;
 - при $dy < 0$ – решение найдено (максимум пройден и значение функции уменьшается).

Алгоритм решения задачи поиска экстремума.

- 5) Задать исходные данные, см. пункты 4,5.
- 6) Начало решения – определить начальные значения:
 - аргумента, $t = 0$;
 - функции, $y = a1 + b1*t - c1*t*t = 0,2 + 4*0 - 1,1*0*0 = 0,2$;
 - номера шага, $N = 0$.
- 7) Решение на шаге:
 - запомнить значение функции на предыдущем шаге $y^P = y$;
 - определить номер шага, $N = N + 1$.
 - определить значение аргумента на шаге $t = t + dt$;
 - определить значение функции на шаге $y = 5 + 2*t - 1,1*t*t$;
 - определить знак приращения функции на шаге $dy = y - y^P$.
- 8) Оценка результата на шаге:
 - при $dy > 0$ – решение не найдено, необходимо делать очередной шаг.
 - при $dy < 0$ – решение найдено (максимум пройден и значение функции уменьшается), необходимо выдать результат.
- 9) Окончание и выдача результата на экран (показать значения):
 - количество шагов N ;
 - аргумента t ;
 - отклика (функции) y ;
 - ошибки dy .

ч3 ЛР 52 “Знакомство с циклом While ” rN3fioWhile

Цель: разработать программную реализацию численного метода.

Исходные условия: знание теории численного метода и наличие студии *NetBeans*.

Постановка задачи (ПЗ). Создать приложение обеспечивающее поиск численным методом максимального значения (экстремума) полинома 2й степени $y = a1 + b1*t -$

$c1 * t * t$, где $a1=0,2$; $b1=4$; $c1=1,1$. *Использовать алгоритм разработанный в лабораторной работе ч2.

Выполнение работы

1) Теория *while* ($n < \text{lim}$) {операторы – изменяют n }. Операторы выполняются пока соблюдается условие

2) Создать приложение:

- имя проекта - *rN3fioWhile*;
- имя формы (тип *JFrame*) - *fWhileFio*
- имя пакета - *pWhileFio*

3) Интерфейс приложения (на форме разместить):

- имя формы;
- кнопку *bStartFio* старта задачи;
- текстовое поле *yMaxFio* для вывода результата.

4) В кнопке *bStartFio* создать событие *MouseClicked* обеспечивающее выполнение постановки задачи. Должно получиться подобное нижеприведенному

```
private void bStartFioMouseClicked(java.awt.event.MouseEvent evt) {
    //исходные значения
    int N = 0;
    double a1 = 0.2, b1 = 4, c1 = 1.1, dt = 0.5;
    double yFio, yP, tFio = 0, dy = 0.001;
    b1 = 100;
    dt = 0.1;
    //начало решения
    yFio = a1 + b1 * tFio - c1 * tFio * tFio;
    //решение на шаге
    while (dy > 0) {
        yP = yFio;
        N = N + 1;
        tFio = tFio + dt;
        yFio = a1 + b1 * tFio - c1 * tFio * tFio;
        dy = yFio - yP;
    }
    yMaxFio.setText(String.format("%.3f",yFio));
    t.setText(String.valueOf(tFio));
    n.setText(String.valueOf(N));
}
```

Дополнительное задание.

- Модифицировать форму для вывода на экран значения *tFio* и метод для решения при *b1* равном номеру студента в журнале.
- Запустить приложение.

После выполнения показать преподавателю.

z9 “While” ИЗ 51 “Оценка точности решения задачи”

Постановка задачи: разработать приложение для поиска экстремума функции с шагом вводимым с экрана и выводом на экран всех изменяемых параметров. Результаты выполнения инд. задания разместить в файле *z9fioWhile.doc*.

Выполнение.

- 1) *За основу взять проект *prn3fioWhile*.
- 2) Создать проект *z9FioW* и форму в нем.
- 3) На форме разместить:
 - панель с надписью «*z9FioW Экстремум полинома*» и в ней показать:
 -
 - кнопку для запуска вычислений;
 - заданное dt ;
 - найденные значения y, t, dy, N .
- 4) В кнопке создать событие обеспечивающее:
 - чтение dt с экрана;
 - вычисление экстремума полинома 2-й степени $y=f(t)$ с параметрами: $a1 = 0.2, b1 = \text{номер студента в списке}, c1 = 1.1$,
 - вывод на экран: y, t, dy, N ;
- 5) Запустить приложение и результат вывести на экран.
- 6) Сделать копию формы, отформатировать ее разместить в файле *z9fioWhile.doc*.
- 7) Выполнить три эксперимента $dt=1; 0,1; 0,001$ результаты представить в таблице и разместить в файле *z9fioWhile.doc*.

В=29, Ченцов

dt	Y	T	dY	N
1	190,6	14	-0,7	14
0,01	191,32	13,3	-0,01	133
0,001	196,336	13,18	-1,5E-6	13183

- 8) В файл *z9fioWhile.doc* добавить копию метода в кнопке.
- 9) Файл переслать преподавателю на *dist*.

6. Документация к разработке приложения

д1 ЛР61. Разработка блок – схемы решения задачи

Цель: получить навыки графического представления последовательности и содержания действий по решению задачи.

Исходные условия: умение разработки компьютерных программ.

Постановка задачи: сформировать блок-схему алгоритма задачи к поиску экстремума полинома 2й степени.

Выполнение работы

Этап 1 – Вербальное (текстовое) описание задачи.

Исходные данные – численным методом (последовательного приближения) найти максимум (экстремум) полинома 2й степени $y = a_1 + b_1 * t - c_1 * t * t$, где $a_1 = 0,2$; $b_1 = 4$; $c_1 = 1,1$.

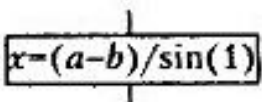
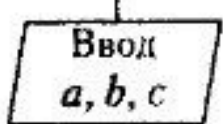

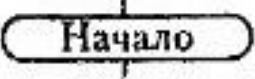
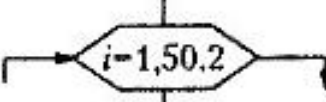
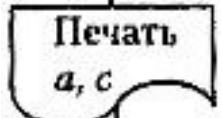
Решение задачи – начиная от y при $t = 0$ и далее с заданным шагом dt определяем текущее значение y . Сравнивая текущее значение с предыдущим – находим величину приращения dy . Решением задачи (экстремум найден) принять ситуацию, когда приращение dy поменяет знак.

Этап 2 – Теория построение блок схемы алгоритма решения задачи

Алгоритм - это формально описанная вычислительная процедура, получающая исходные данные, называемые так же входом алгоритма или его аргументом и выдающая результат вычислений на выход.

Блок-схема алгоритма — графическое изображение алгоритма в виде связанных между собой с помощью стрелок (линий перехода) и блоков — графических символов, каждый из которых соответствует одному шагу алгоритма. Внутри блока дается описание соответствующего действия.

Пример символов блок-схемы

	Вычислительное действие		Ввод-вывод
	Проверка условий		Начало, конец алгоритма
	Цикл FOR		Вывод результатов

Этап 3 - Построение блок-схемы алгоритма решения задачи

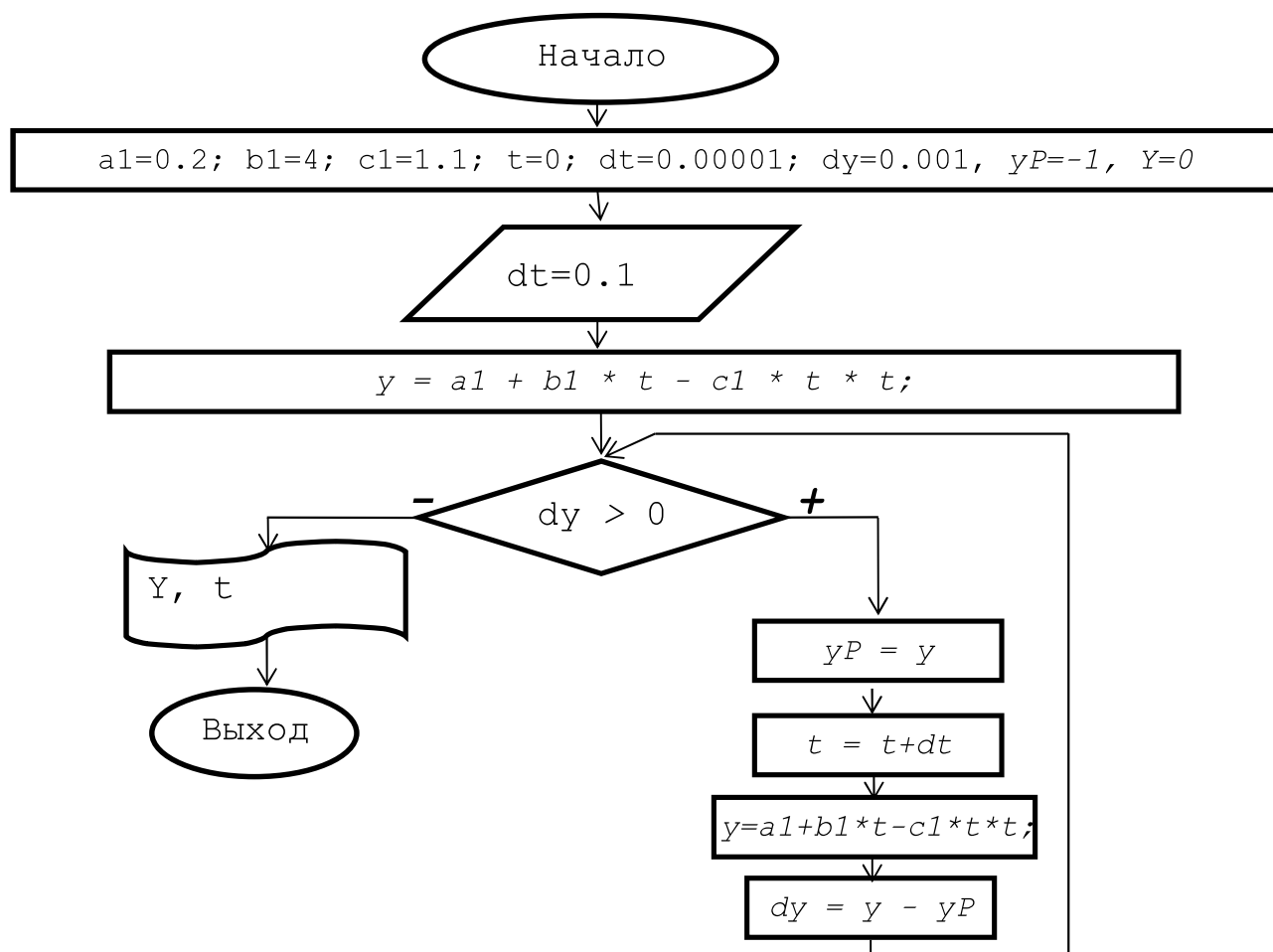


Рисунок. Блок-схема алгоритма поиска экстремума

Дополнительное задание.

- На блок – схеме дополнительно показать: операторы для определения количества выполненных шагов приближения; вывод на печать количества шагов и точности решения.

После выполнения показать преподавателю.

z10 “Блок – схема” ИЗ 61 “Блок – схема к определению Totk”

Постановка задачи: разработать блок – схему задачи по определению даты отказа *totk* детали как решение системы уравнений:

$Y = f(t)$ - функция изменения технического состояния детали;

$Y_{np} = f(t)$ - функция изменения предельного технического состояния детали.

Функция $Y = f(t)$ - представлена точками приведенными в таблице индивидуального задания z5 «Таблица».

Функция $Y_{np} = f(t)$ описывается полиномом 1й степени $Y_{np} = Y_{n-s} * t$ (линейная функция).

Результаты выполнения инд. задания разместить в файле *z10fioBS.doc*.

Выполнение.

- 1) В файл *z10fioBS.doc* **вставить** таблицу *DiagnostFIO* из *z5 «Таблица»*.
- 2) Выполнить параметризацию функции $Y_{np} = f(t)$. Используя данные таблицы из *z5* определить:
 - Y_H – значение технического состояния при последнем t ;
 - $s = b * Y_H / tk$, где tk - максимальное значение t в таблице; b - поправочный коэффициент принять с учетом последней цифры номера студента в журнале преподавателя: (0 – (b=0,82); 1- (b=0,56); 2-(b=0,78); 3-(b=0,62); 4-(d=0,84); 5 – (b=0,6); 6 – 0,7; 7-0,73; 8-0,78;9-0,8).
 - Выражение функции Y_{np} **привести** в файле *z10fioBS.doc*
- 3) Сроком отказа $tomk$ детали считать решение системы уравнений $Y = f(t)$ и $Y_{np} = f(t)$ когда $Y \geq Y_{np}$.
- 4) Решить систему, используя графический метод (см. ч1 ЛР 51 “Использование численного метода”). В файл *z10fioBS.doc* **добавить копию** графического решения.
- 5) Выполнить контрольный пример с использованием численного решения системы (см. ч1 ЛР 51 “Использование численного метода”). В файл *z10fioBS.doc* **добавить копию** контрольный пример.
- 6) Разработать блок – схему численного решения системы и **добавить** в файл *z10fioBS.doc*.
- 7) Файл переслать преподавателю на *dist*.

z11 “ПриложениеТотк” ИЗ 62 “Определение даты отказа”

Постановка задачи: используя блок-схему разработанную в *z10 “Блок – схема”* разработать приложение к определению срока отказа $tomk$ детали. Результаты выполнения инд. задания разместить в файле *z11fioTotk.doc*.

Выполнение.

- 1) Создать проект *z11Fio* взяв за основу приложение, разработанное в *z8 “Средняя V”* с подключенной таблицей *DiagnostFIO*.
- 2) Требования к интерфейсу- показать результаты: количество шагов N ; значения t ; Y и ошибку dY .
- 3) В методе, решающем систему уравнений использовать цикл *while* и движение по точкам таблицы *DiagnostFIO*.
- 4) Разработать приложение и запустить его.
- 5) Сделать копию формы с решением, отформатировать ее и **добавить** в файл *iz62fio.doc*.
- 6) В файл *iz62fio.doc* **добавить копию** текста метода решения системы уравнений к определению срока отказа $tomk$.
- 7) Файл переслать преподавателю на *dist*.

д2 ЛР62. Комплект документов к решению задачи

Цель: получить представление о комплекте документов, разрабатываемых к решению задачи.

Исходные условия: знания, полученные при выполнении лабораторных работ, связанных с поиском экстремума функции.

Постановка задачи: сформировать комплект документов к разработке приложения используемого для поиска экстремума функции.

Выполнение работы

Этап 1 – Вербальное (текстовое) описание задачи.

Характеристика задачи – численным методом (последовательного приближения) найти максимум (экстремум) полинома 2й степени $y = a1 + b1*t - c1*t*t$.

Исходные данные – параметры полинома задаются в тексте приложения: $a1=0,2$; $b1=4$; $c1=1,1$. *База данных не используется.

Решение задачи – начиная от y при $t=0$ и далее с заданным шагом dt определяем текущее значение y . Сравнивая текущее значение с предыдущим – находим величину приращения dy . Решением задачи (экстремум найден) принять ситуацию, когда приращение dy поменяет знак.

Использование приложения – в запущенном приложении обеспечить изменение с экрана требуемой точности решения (изменяя dt), запуск задачи и вывод результатов: y , t , точность решения и количество шагов.

Этап 2. Графический метод решения задачи поиска экстремума.

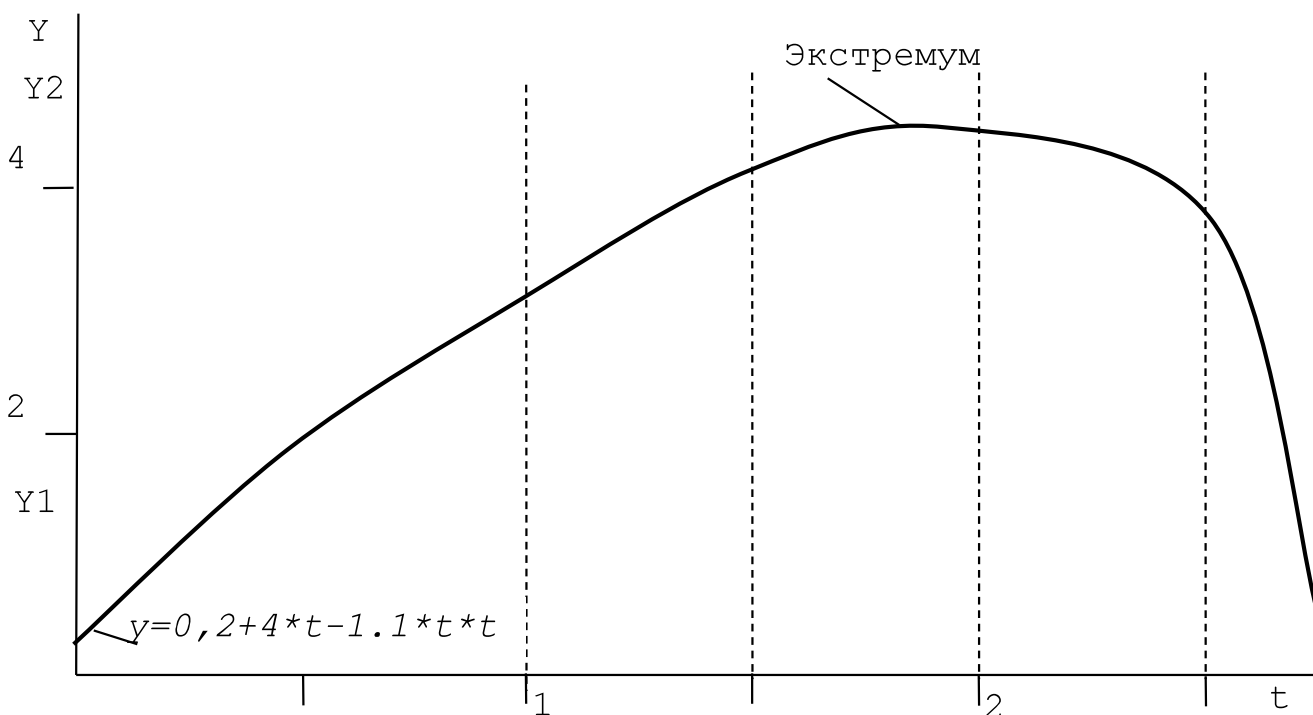


Рисунок 1. Графическое решение поиска экстремума.

Этап 3. Алгоритм решения задачи поиска экстремума.

- 1) Задать исходные данные, см. этап 1.
- 2) Начало решения – определить начальные значения:
 - аргумента, $t=0$;
 - функции, $y = a1 + b1*t - c1*t*t = 0,2 + 4*0 - 1.1*0*0 = 0,2$;
 - номера шага, $N=0$.
- 3) Решение на шаге:
 - запомнить значение функции на предыдущем шаге $y^P = y$;
 - определить номер шага, $N=N+1$.
 - определить значение аргумента на шаге $t=t + dt$;
 - определить значение функции на шаге $y = 5 + 2*t - 1,2*t*t$
 - определить знак приращения функции на шаге $dy = y - y^P$.
- 4) Оценка результата на шаге:
 - при $dy > 0$ – решение не найдено, необходимо делать очередной шаг.
 - при $dy < 0$ – решение найдено (максимум пройден и значение функции уменьшается), необходимо выдать результат.
- 5) Окончание и выдача результата на экран (показать значения):
 - количество шагов N ;
 - аргумента t ;
 - отклика (функции) y ;
 - ошибки dy .

Этап 4 – Построение блок схемы алгоритма решения задачи

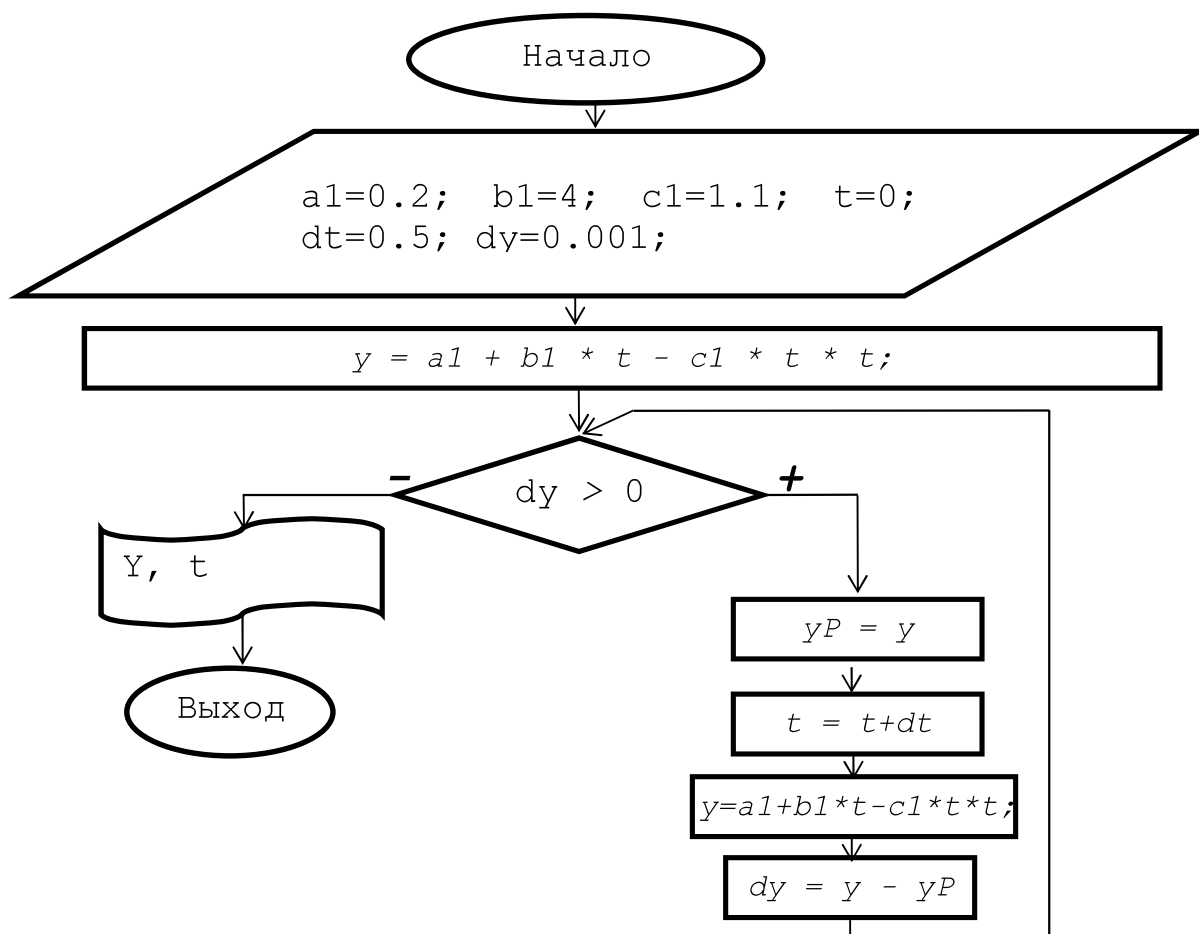


Рисунок 2. Блок схема алгоритма поиска экстремума

Этап 5. Контрольный пример.

1) Начало решения – определить начальные значения:

- номера шага, $N=0$.
- аргумента, $t=0$;
- функции, $y_0 = a1 + b1*t - c1*t*t = 0,2 + 4*0 - 1.1*0*0 = 0,2$;

2) Шаг первый $N=1$ численного решения:

- запомнить значение функции на предыдущем шаге $y^P = y = 0.2$;
- определить номера шага, $N = N+1 = 0+1 = 1$;
- определить значение аргумента, $t = t + dt = 0 + 0.5 = 0.5$;
- определить значение функции, $y = 0,2 + 4*0.5 - 1.1*0.5*0.5 = 1.925$;
- ошибка решения $dy = y - y^P = 1.925 - 0.2 = 1.725$
- принять решение - при $dy > 0$ – решение не найдено, делать следующий шаг.

3) Шаг второй $N=2$ и последующие – результаты по всем шагам приведены в таблице:

N	t	y	Dy
0	0	0,2	

1	0,5	1,925	1,725
2	1	3,1	1,175
3	1,5	3,725	0,625
4	2	3,8	0,075
5	2,5	3,325	-0,475

4) Результат решения – данные оптимума функции:

- количество шагов $N=5$.
- значение аргумента $t= 2.5$;
- значение функции $y = 3.325$;
- ошибка -0.475 .

Этап 6. Программная реализация алгоритма

1) Программная реализация разработана в лаб.работе 51.

2) Имена компонент проекта

- имя проекта - *Lr51WhileChen*;
- имя формы (тип *JFrame*) - *fWhileChen*
- имя пакета - *pWhileChen*

3) Интерфейс приложения (на форме разместить):

- имя формы;
- кнопку *bStartChen* старта задачи;
- текстовое поле *yMaxChen* для вывода результата.

4) В кнопке *bStartChen* создать событие *MouseClicked* обеспечивающее выполнение постановки задачи. Должно получиться подобное нижеприведенному

```
private void bStartChenMouseClicked(java.awt.event.MouseEvent evt) {
    //исходные значения
    int N = 0;
    double a1 = 0.2, b1 = 4, c1 = 1.1, dt = 0.5;
    double yChen, yP, tChen = 0, dy = 0.001;
    //начало решения
    yChen = a1 + b1 * tChen - c1 * tChen * tChen;
    //решение на шаге
    while (dy > 0) {
        yP = yChen;
        N = N + 1;
        tChen = tChen + dt;
        yChen = a1 + b1 * tChen - c1 * tChen * tChen;
        dy = yChen - yP;
    }
    yMaxChen.setText(String.valueOf(yChen));
}
```

Этап 7. Инструкция пользователю

1) Ограничение. Приложение обеспечивает поиск экстремума только одного полинома $y=0.2+4*t-1.1*t*y$.

- 2) Задать исходные данные – ввести с экрана:
 - dt , шаг по аргументу;
- 3) Запустить задачу на выполнение – нажать кнопку Старт.
- 4) Получить результат решения с экрана:
 - y – значение функции в точке экстремума;
 - t – значение аргумента в точке экстремума;
 - dy , шаг по аргументу;
 - N – количество шагов при поиске экстремума.

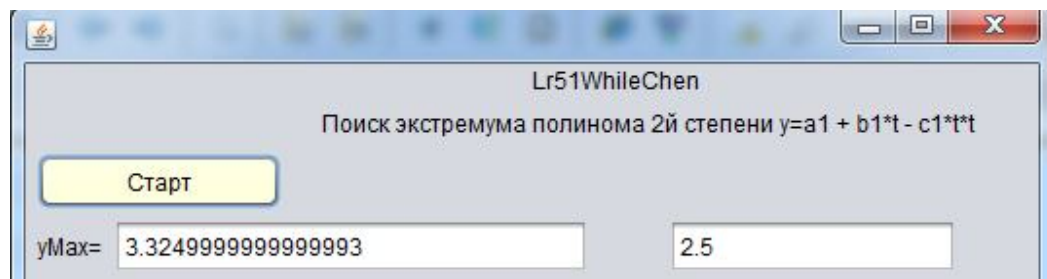


Рисунок 3. Форма (экран) приложения.

- 5) Тестовая задача при $dt=0.5$, результаты решения:
 - $y= 3.325$, значение функции в точке экстремума;
 - $t= 2.5$, значение аргумента в точке экстремума;
 - $dy =$, шаг по аргументу;
 - $N =$, количество шагов при поиске экстремума.

Дополнительное задание.

- На блок – схеме дополнительно показать: определение количества выполненных шагов приближения; вывод на печать количества шагов и точности решения.

После выполнения показать преподавателю.

7. Зачетная работа

ZR “Разработать комплект документов к определению экстремума функции”

Постановка задачи: Разработать комплект документов к решению задачи поиска экстремума функции описанной полиномом 3й степени.

Выполнение.

- 1) Выполнить зачетную работу в соответствии со структурой и содержанием примера зачетной работы для варианта 77.

***В примере зачетной работы намеренно введены несоответствия, которые должны быть устранены в работе студента.**
- 2) Данные своего варианта (значения параметров полинома) взять из таблицы, где номер варианта – номер студента в списке преподавателя.

Таблица Параметры к зачетной работе

Цифра варианта	Все варианты		Параметры по 1й цифре		Параметры по 2й цифре	
	t0	dt	a	b	c	d
0	0	0,50	0,20	1,80	1,05	0,50
1	0	0,50	0,25	1,70	1,10	0,60
2	0	0,50	0,25	3,40	1,20	0,50
3	0	0,50	1,20	3,40	1,20	0,50
4	0	0,50	1,10	4,20	0,90	0,50
5	0	0,50	0,80	3,90	1,80	0,70
6	0	0,50	0,80	2,90	1,80	0,70
7	0	0,50	0,25	2,90	1,80	0,70
8	0	0,50	1,30	3,80	1,80	0,70
9	0	0,50	1,50	4,20	2,10	0,80