



Некоммерческое
акционерное
общество

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И
СВЯЗИ**

Кафедра
компьютерной
технологий

СОВРЕМЕННЫЕ МЕТОДЫ И СРЕДСТВА ПРОГРАММИРОВАНИЯ

Методические указания к лабораторным работам
для студентов специальности
5В070400 – Вычислительная техника и программное обеспечение

Алматы 2015

СОСТАВИТЕЛИ: Е.С.Турганбаев., М.И. Кожамбердиева. Современные методы и средства программирования. Методические указания к лабораторным работам для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2014. – 46 с.

В методических указаниях представлены описания лабораторных работ по курсу «Современные методы и средства программирования», выполнение которых определяет фундаментальную основу профессионального образования по первой части курса «Современные методы и средства программирования», Они составлены в соответствии с требованиями квалификационной характеристики специалиста, педагогико-психологических основ организации и проведения лабораторных занятий. Предлагаемые лабораторные работы выполняются как посредством организации практических экспериментов на языке Java, так и путем проведения виртуальных исследований с помощью системы в программной среде IDE Eclipse.

Современные методы и средства программирования. Методические указания предназначены для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение.

Ил.33, табл. 2, библиогр.-5 назв.

Рецензент: доцент А.С.Байкенов.

Печатается по плану издания некоммерческого акционерного общества «Алматинский университет энергетики и связи» на 2014 г.

© НАО «Алматинский университет энергетики и связи», 2015 г

Содержание

| | |
|--|----|
| Введение | 4 |
| 1 Лабораторная работа № 1. Простое консольное приложение java..... | 5 |
| 1.1 Методические указания к выполнению работы | 5 |
| 1.2 Порядок выполнения рабочего задания..... | 9 |
| 1.3 Индивидуальные задания | 13 |
| 1.4 Варианты задания..... | 14 |
| 2 Лабораторная работа № 2. Java инкапсуляция..... | 16 |
| 2.1 Порядок выполнения рабочего задания..... | 16 |
| 2.2 Индивидуальные задания | 18 |
| 2.3 Варианты задания | 18 |
| 2.4 Постановка задачи | 21 |
| 2.5 Дополнительные вопросы и задания..... | 22 |
| 3 Лабораторная работа № 3. Java наследование | 22 |
| 3.1 Порядок выполнения рабочего задания..... | 22 |
| 3.2 Индивидуальные задания | 24 |
| 3.3 Варианты задания | 25 |
| 3.4 Дополнительные вопросы и задания..... | 32 |
| 4 Лабораторная работа № 4. Java массивы интерфейс comparable | 32 |
| 4.1 Порядок выполнения рабочего задания..... | 32 |
| 4.2 Индивидуальные задания | 36 |
| 4.3 Варианты задания | 36 |
| 4.4 Дополнительные вопросы и задания..... | 44 |
| Список литературы | 45 |

Ведение

В предлагаемом в методическом руководстве к лабораторным занятиям представлены два типа рабочих заданий, отличающихся по характеру выполнения, и объединенных в соответствующие подразделы. Задания первого типа выполняются посредством проведения практических измерений и наблюдений на «Современные методы и средства программирования», а второй тип заданий организовывается в виде виртуальных экспериментов, выполняемых посредством моделирования исследуемых устройств в программной среде IDE Eclipse. При этом часть заданий второго типа выносятся в качестве задания для самостоятельной работы студентов (СРС). Предлагаемая методика организации лабораторных занятий проверена многолетней практикой преподавания автора и испытана временем.

Различные формы организации учебного исследования обеспечивают более глубокие знания студентов по теории изучаемых элементов и практических навыков по проведению практических и моделирующих экспериментов.

Выполнения учебных исследований как в виде практических, так и в виде виртуальных экспериментов создает возможность для организации лабораторных занятий в случае недостаточной обеспеченности лабораторным оборудованием. В таком случае одной половине студентов учебной группы можно предложить выполнение практической части эксперимента, а другой – выполнить виртуальные исследования, а затем им поменять учебные задания.

Предлагаемые в методическом руководстве задания по выполнению виртуальных исследований с помощью системы IDE Eclipse в достаточном объеме обеспечивают выполнение целей и задач лабораторных занятий, что создает условия и для самостоятельного применения их в случае учебно-производственной необходимости (например, в случае отсутствия лабораторного оборудования).

По выполненному учебному исследованию студенты должны подготовить отчет. Результаты выполнения задания на СРС и краткие ответы на вопросы.

1 Лабораторная работа №1. Простое консольное приложение Java

Цель работы:

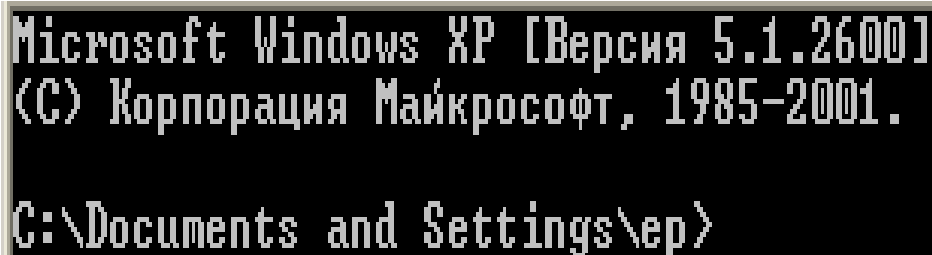
- ознакомиться со структурой простого консольного приложения Java;
- научиться компилировать исходный Java код в байт код с командной строки;
- научиться запускать байт код с командной строки;
- ознакомиться с IDE Eclipse;
- ознакомиться с консольным вводом\выводом в Java;
- ознакомиться с преобразованием строковых и числовых данных в Java;
- ознакомиться с синтаксисом операторов управления в Java.

1.1 Методические указания к выполнению работы

1.1.1 Создать первое приложение «Hello world» с помощью блокнота.

1.1.2 Запустить cmd.

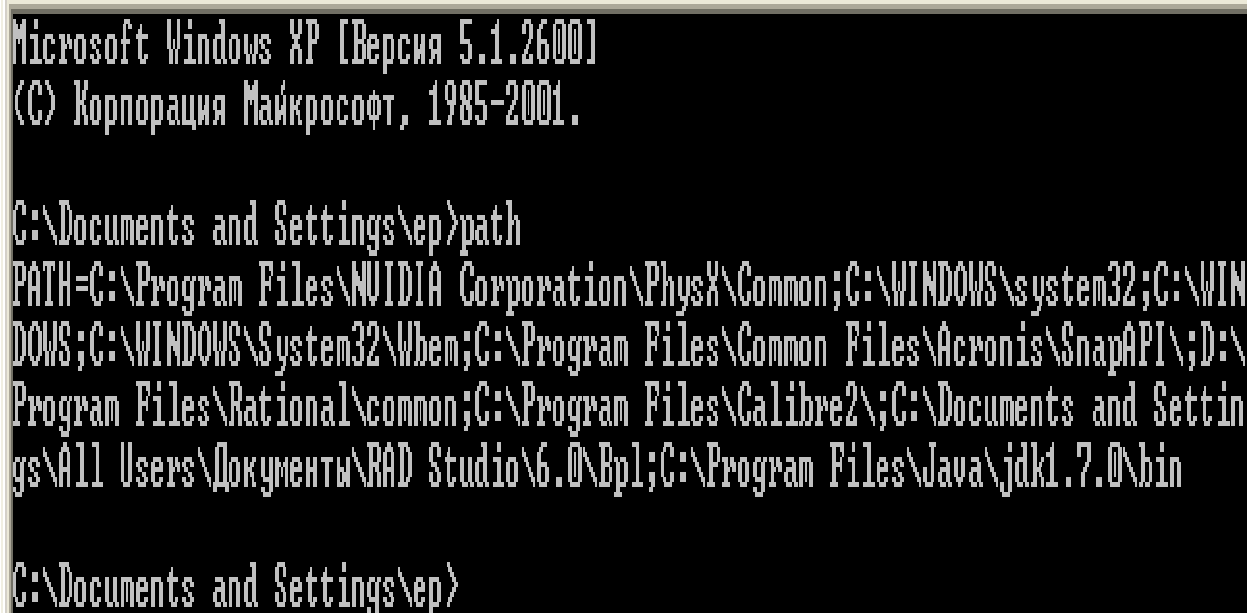
1.1.3



```
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\ep>
```

Рисунок 1-Проверяем установки переменной среды Path



```
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\ep>path
PATH=C:\Program Files\NVIDIA Corporation\PhysX\Common;C:\WINDOWS\system32;C:\WIN
DOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\Common Files\Acronis\SnapAPI\;D:\
Program Files\Rational\common;C:\Program Files\Calibre2\;C:\Documents and Settin
gs\All Users\Документы\RAD Studio\6.0\Bpl;C:\Program Files\Java\jdk1.7.0\bin

C:\Documents and Settings\ep>
```

Рисунок 1.2 - Проверяем наличие установленного компилятора Javac

```

C:\Documents and Settings\ep>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:<lines,vars,source> Generate only some debugging info
  -nowarn           Generate no warnings
  -verbose          Output messages about what the compiler is doing
  -deprecation      Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotation processors
  -cp <path>        Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>   Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:<none,only> Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>    Specify where to place generated class files
  -s <directory>    Specify where to place generated source files
  -implicit:<none,class> Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release> Provide source compatibility with specified release

  -target <release> Generate class files for specific VM version
  -version           Version information
  -help             Print a synopsis of standard options
  -Akey[=value]     Options to pass to annotation processors
  -X                Print a synopsis of nonstandard options
  -J<flag>          Pass <flag> directly to the runtime system
  -Werror           Terminate compilation if warnings occur
  @<filename>       Read options and filenames from file

C:\Documents and Settings\ep>

```

Рисунок 1.3 - Переходим в директорию, предназначенную для размещения проектов Java

```
C:\Documents and Settings\ep>d:
D:\>cd java
D:\Java>dir
Том в устройстве D имеет метку DATA-NEW
Серийный номер тома: B762-FCE5

Содержимое папки D:\Java

16.06.2013  14:05    <DIR>          .
16.06.2013  14:05    <DIR>          ..
12.06.2013  12:56    <DIR>          .metadata
              0 файлов              0 байт
              3 папок   124 413 612 032 байт свободно

D:\Java>
```

Рисунок 1.4 - С помощью блокнота создаем файл hello.txt

```
public class hello {
    public static void main(String[] args) {
        System.out.println("PRIVET");
    }
}
```

Рисунок 1.5 - Переименовываем файл hello.txt в hello.java

```
C:\WINDOWS\system32\cmd.exe
D:\Java>rename
Ошибка в синтаксисе команды.

D:\Java>dir
Том в устройстве D имеет метку DATA-NEW
Серийный номер тома: B762-FCE5

Содержимое папки D:\Java

17.06.2013  06:59    <DIR>          .
17.06.2013  06:59    <DIR>          ..
12.06.2013  12:56    <DIR>          .metadata
17.06.2013  06:59                117 hello.txt
                1 файл(ов)                117 байт
                3 папок  124 413 612 032 байт свободно

D:\Java>rename hello.txt hello.java

D:\Java>dir
Том в устройстве D имеет метку DATA-NEW
Серийный номер тома: B762-FCE5

Содержимое папки D:\Java

17.06.2013  07:04    <DIR>          .
17.06.2013  07:04    <DIR>          ..
12.06.2013  12:56    <DIR>          .metadata
17.06.2013  06:59                117 hello.java
                1 файл(ов)                117 байт
                3 папок  124 413 612 032 байт свободно

D:\Java>
```

Рисунок 1.6 - Компилируем файл hello.java в байт код


```
D:\Java>javac hello.java
D:\Java>
```

```
D:\Java>javac hello.java
D:\Java>java hello
PRIVET
D:\Java>
```

Рисунок 1.7 - Выполняем байт код

Рисунок 1.8 - Выполняем байт код

1.2 Порядок выполнения рабочего задания

1.2.1 Создать первое приложение «Hello world» с помощью IDE Eclipse.

1.2.2 Запустить IDE Eclipse.

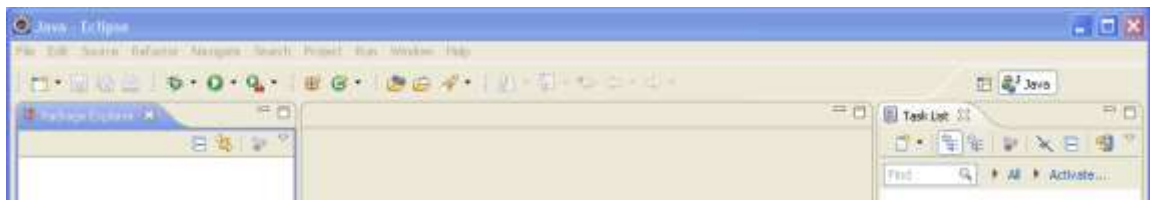


Рисунок 1.9 - Создать новый проект java с именем hello1

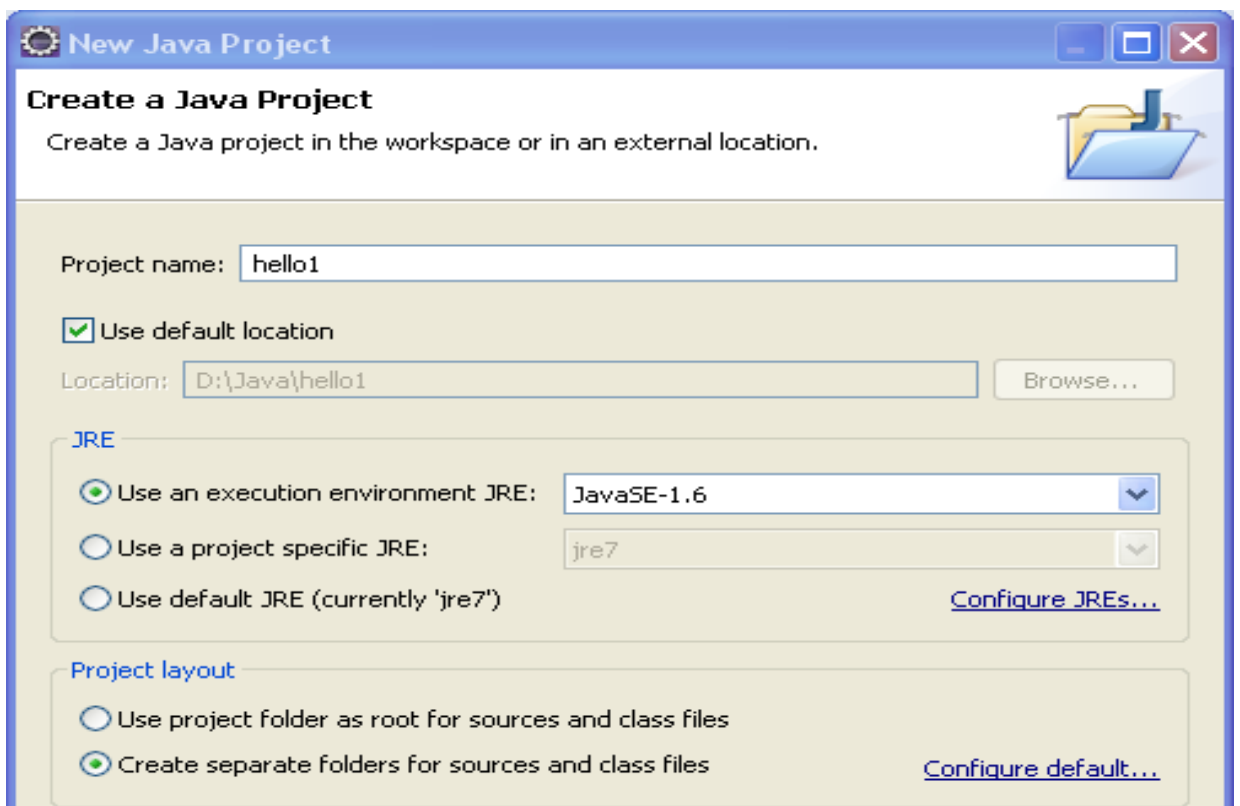


Рисунок 1.10 - Создать новый проект java с именем Hello1

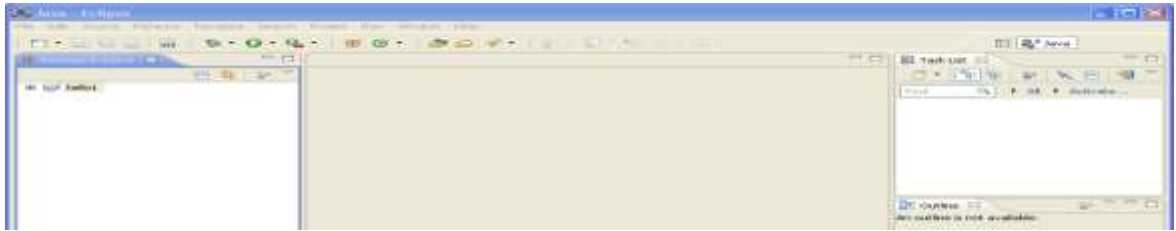


Рисунок 1.11 - В проекте Hello1 создать класс hellotest

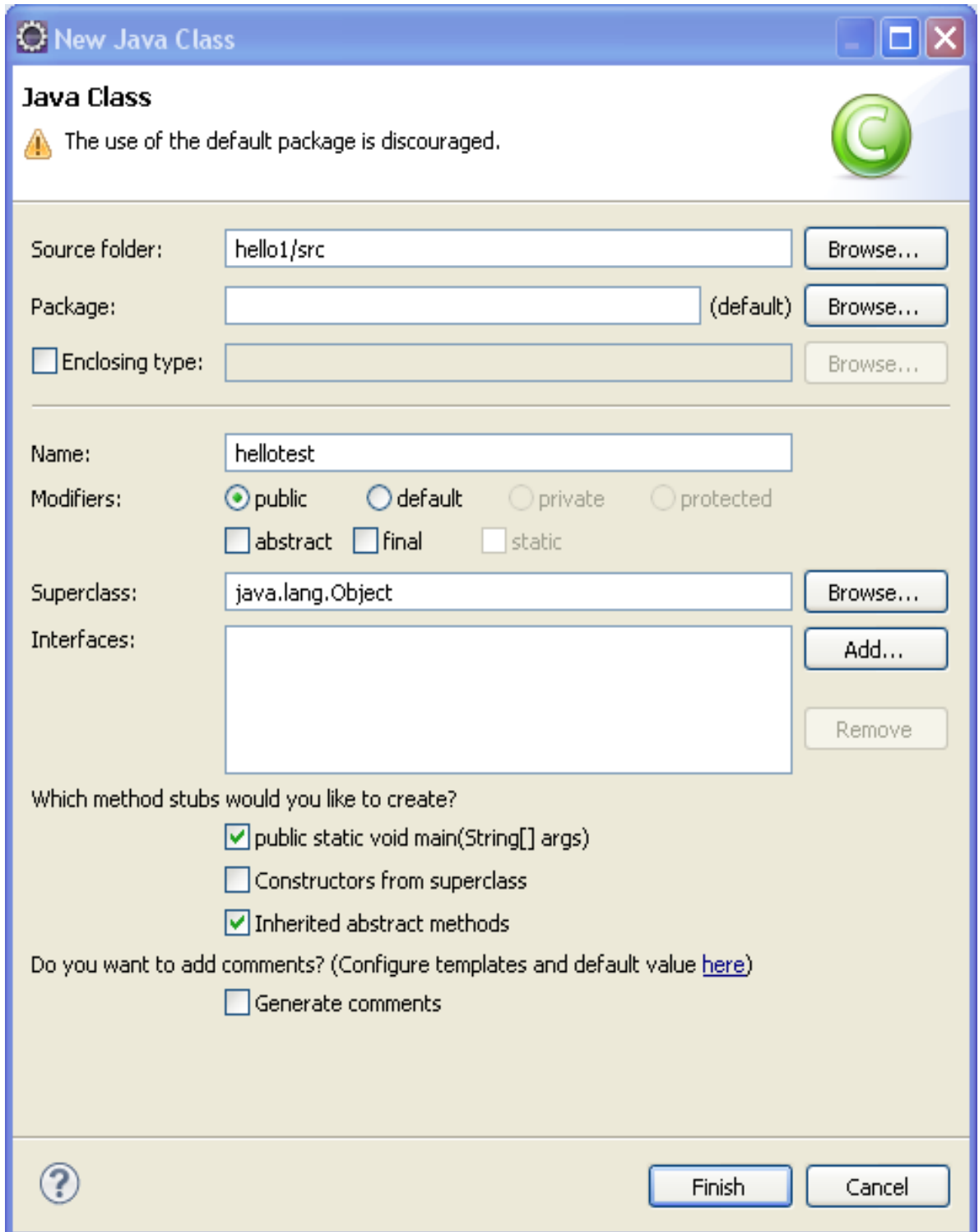


Рисунок 1.12 - В проекте hello1 создать класс hellotest

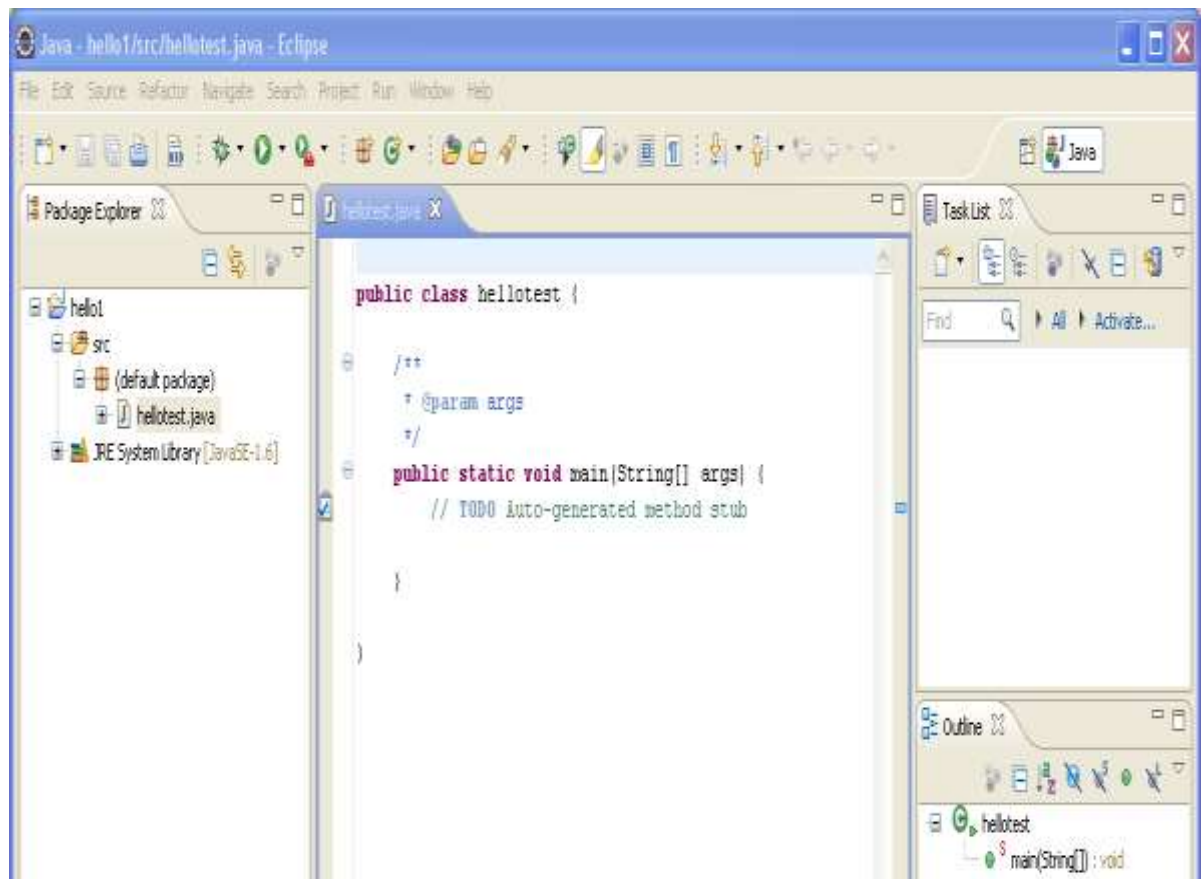


Рисунок 1.13 - Войти в код

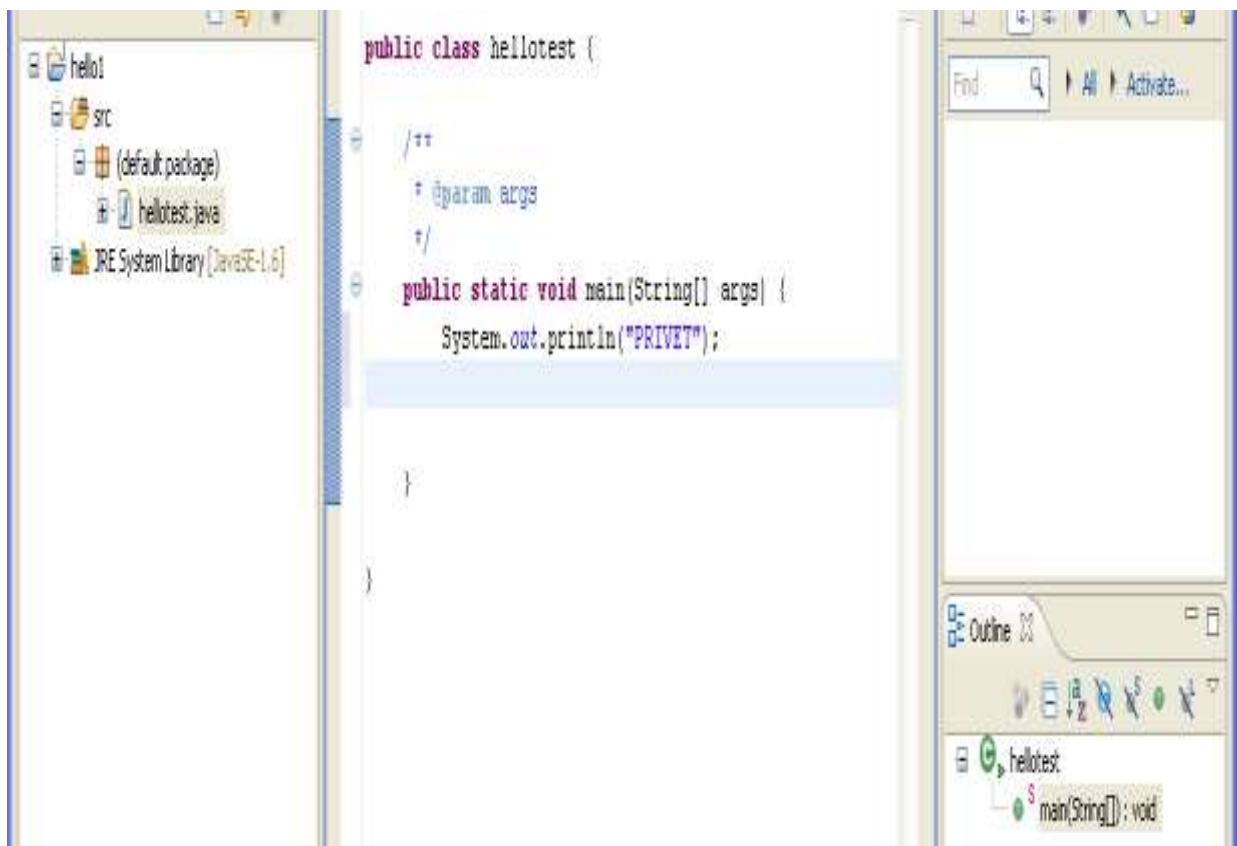


Рисунок 1.14 - Компилируем в байт код и запускаем байт код в IDE Eclipse

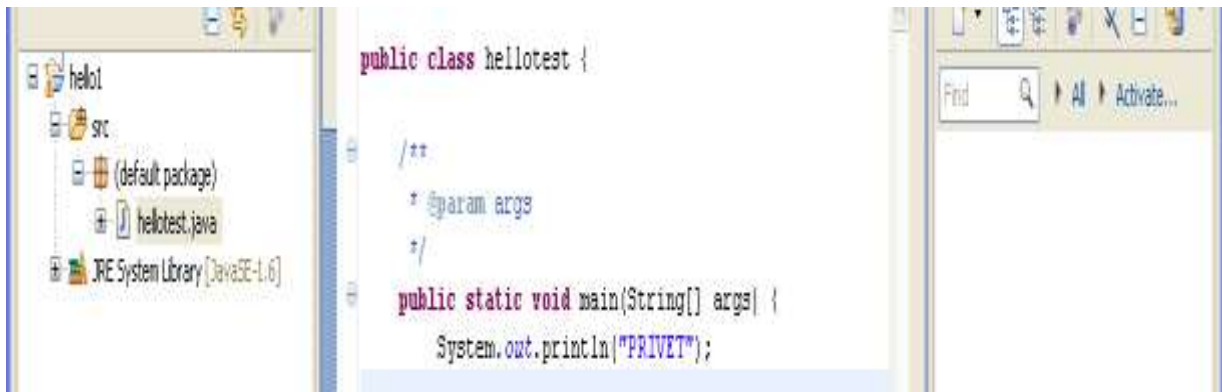


Рисунок 1.15 - Запускаем байт код в IDE Eclipse

1.2.3 Запустить в командной строке приложение, созданное с помощью IDE Eclipse.

1.2.4 В режиме командной строки переходим в директорию, созданную IDE Eclipse.

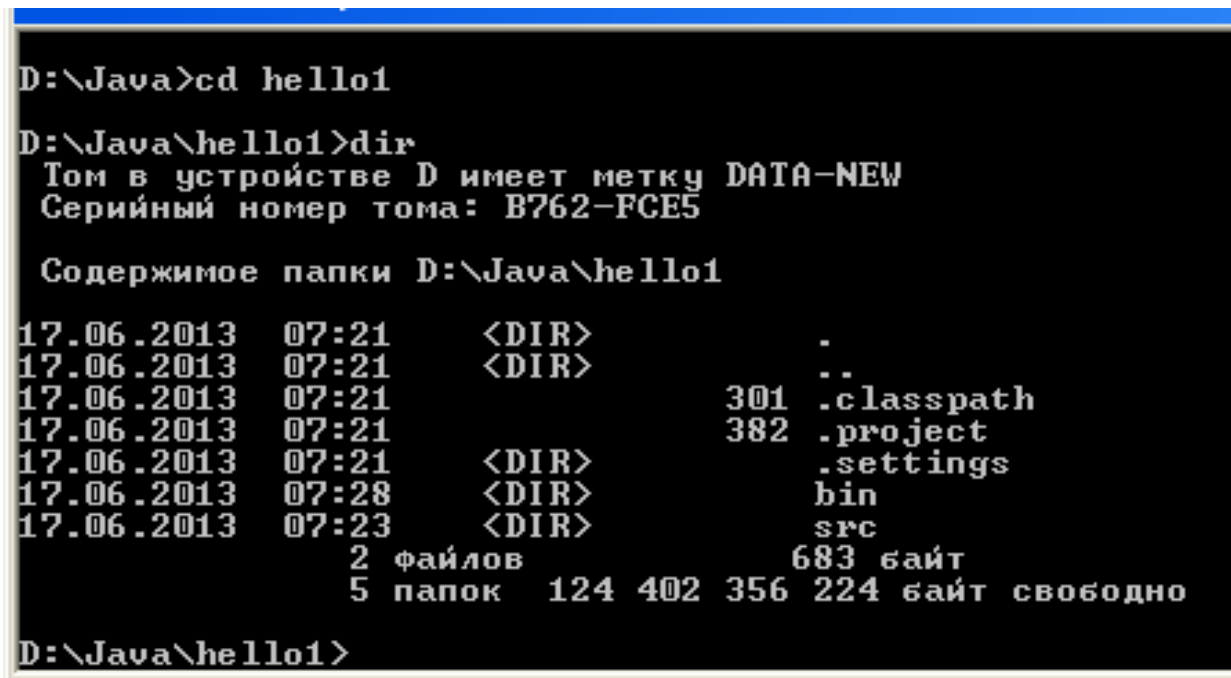


Рисунок 1.16 - Созданную IDE Eclipse

1.2.5 Запускаем байт код, созданный с помощью IDE Eclipse в командной строке.



Рисунок 1.17 - IDE Eclipse в командной строке

1.2.6 Консольный ввод/вывод. Функции преобразования строк и чисел. Провести вычисления по формуле: $c=a+b$, параметры вводятся с консоли.

Текст программы.

```
import java.util.*;

public abstract class calc
{
    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);
        int a=sc.nextInt();
        System.out.println(a);
        sc.close();
    }
}
```

1.3 Индивидуальные задания (Вариант заданий выбирается по номеру в журнале у преподавателя)

Общая часть.

Для своего варианта вычислить сумму числового ряда.

Таблица 1- Варианты задания

| Вариант задания | Вариант по Рябушко | Вариант задания | Вариант по Рябушко |
|-----------------|--------------------|-----------------|--------------------|
| 1 | 5 | 31 | 9 |
| 2 | 6 | 32 | 10 |
| 3 | 7 | 33 | 11 |
| 4 | 8 | 34 | 12 |
| 5 | 9 | 35 | 13 |
| 6 | 10 | 36 | 14 |
| 7 | 11 | 37 | 15 |
| 8 | 12 | 38 | 16 |
| 9 | 13 | 39 | 17 |
| 10 | 14 | 40 | 18 |
| 11 | 15 | 41 | 19 |
| 12 | 16 | 42 | 20 |
| 13 | 17 | 43 | 21 |
| 14 | 18 | 44 | 22 |
| 15 | 19 | 45 | 23 |
| 16 | 20 | 46 | 24 |
| 17 | 21 | 47 | 25 |

| | | | |
|----|----|----|----|
| 18 | 22 | 48 | 26 |
| 19 | 23 | 49 | 27 |
| 20 | 24 | 50 | 28 |
| 21 | 25 | 51 | 29 |
| 22 | 26 | 52 | 30 |
| 23 | 27 | 53 | 1 |
| 24 | 28 | 54 | 2 |
| 25 | 29 | 55 | 3 |
| 26 | 30 | 56 | 4 |
| 27 | 1 | 57 | 5 |
| 28 | 2 | 58 | 6 |
| 29 | 3 | 59 | 7 |
| 30 | 4 | 60 | 8 |
| 61 | 9 | 63 | 11 |
| 62 | 10 | 64 | 12 |
| 65 | 13 | 66 | 14 |

1.4 Варианты задания

- 1) $\sum_{n=1}^{\infty} \frac{1}{n(n+2)}$ · (Ответ: $S = \frac{3}{4}$.);
- 2) $\sum_{n=1}^{\infty} \frac{3^n + 4^n}{12^n}$ · (Ответ: $S = \frac{5}{6}$.);
- 3) $\sum_{n=0}^{\infty} \frac{1}{(2n+5)(2n+7)}$ · (Ответ: $S = \frac{1}{10}$.);
- 4) $\sum_{n=1}^{\infty} \frac{2^n + 5^n}{10^n}$ · (Ответ: $S = \frac{5}{4}$.);
- 5) $\sum_{n=0}^{\infty} \frac{1}{(n+5)(n+6)}$ · (Ответ: $S = \frac{1}{5}$.);
- 6) $\sum_{n=1}^{\infty} \frac{5^n - 2^n}{10^n}$ · (Ответ: $S = \frac{3}{4}$.);
- 7) $\sum_{n=0}^{\infty} \frac{1}{(2n+7)(2n+9)}$ · (Ответ: $S = \frac{1}{14}$.);
- 8) $\sum_{n=1}^{\infty} \frac{4^n - 3^n}{12^n}$ · (Ответ: $S = \frac{1}{6}$.);
- 9) $\sum_{n=1}^{\infty} \frac{1}{(n+6)(n+7)}$ · (Ответ: $S = \frac{1}{7}$.);

- 10) $\sum_{n=1}^{\infty} \frac{3^n + 5^n}{15^n} \cdot$ (Ответ: $S = \frac{3}{4} \cdot$);
- 11) $\sum_{n=1}^{\infty} \frac{1}{(n+9)(n+10)} \cdot$ (Ответ: $S = \frac{1}{10} \cdot$);
- 12) $\sum_{n=1}^{\infty} \frac{5^n - 3^n}{15^n} \cdot$ (Ответ: $S = \frac{1}{4} \cdot$);
- 13) $\sum_{n=1}^{\infty} \frac{1}{(n+7)(n+8)} \cdot$ (Ответ: $S = \frac{1}{8} \cdot$);
- 14) $\sum_{n=1}^{\infty} \frac{2^n + 7^n}{14^n} \cdot$ (Ответ: $S = \frac{7}{6} \cdot$);
- 15) $\sum_{n=0}^{\infty} \frac{1}{(n+2)(n+3)} \cdot$ (Ответ: $S = \frac{1}{2} \cdot$);
- 16) $\sum_{n=1}^{\infty} \frac{7^n - 2^n}{14^n} \cdot$ (Ответ: $S = \frac{5}{6} \cdot$);
- 17) $\sum_{n=0}^{\infty} \frac{1}{(n+3)(n+4)} \cdot$ (Ответ: $S = \frac{1}{3} \cdot$);
- 18) $\sum_{n=1}^{\infty} \frac{4^n + 5^n}{20^n} \cdot$ (Ответ: $S = \frac{7}{12} \cdot$);
- 19) $\sum_{n=1}^{\infty} \frac{1}{(n+4)(n+5)} \cdot$ (Ответ: $S = \frac{1}{5} \cdot$);
- 20) $\sum_{n=1}^{\infty} \frac{5^n - 4^n}{20^n} \cdot$ (Ответ: $S = \frac{1}{12} \cdot$);
- 21) $\sum_{n=0}^{\infty} \frac{1}{(2n+1)(2n+3)} \cdot$ (Ответ: $S = \frac{1}{2} \cdot$);
- 22) $\sum_{n=1}^{\infty} \frac{7^n + 3^n}{21^n} \cdot$ (Ответ: $S = \frac{2}{3} \cdot$);
- 23) $\sum_{n=0}^{\infty} \frac{1}{(2n+3)(2n+5)} \cdot$ (Ответ: $S = \frac{1}{6} \cdot$);
- 24) $\sum_{n=1}^{\infty} \frac{7^n - 3^n}{21^n} \cdot$ (Ответ: $S = \frac{1}{3} \cdot$);
- 25) $\sum_{n=1}^{\infty} \frac{1}{(3n-1)(3n+2)} \cdot$ (Ответ: $S = \frac{1}{6} \cdot$);
- 26) $\sum_{n=1}^{\infty} \frac{3^n + 8^n}{24^n} \cdot$ (Ответ: $S = \frac{9}{14} \cdot$);

$$27) \sum_{n=1}^{\infty} \frac{1}{(3n+1)(3n+4)}. \quad (\text{Ответ: } S = \frac{1}{12}.);$$

$$28) \sum_{n=1}^{\infty} \frac{8^n - 3^n}{24^n}. \quad (\text{Ответ: } S = \frac{5}{14}.);$$

$$29) \sum_{n=1}^{\infty} \frac{1}{(3n+2)(3n+5)}. \quad (\text{Ответ: } S = \frac{1}{15}.);$$

$$30) \sum_{n=1}^{\infty} \frac{9^n - 2^n}{18^n}. \quad (\text{Ответ: } S = \frac{7}{8}.).$$

2 Лабораторная работа №2. Java, инкапсуляция

Цель работы:

- ознакомиться с использованием принципа инкапсуляции в Java;
- конструкторы;
- модификаторы доступа;
- статические члены.

2.1 Порядок выполнения работы

Создать класс «Собака», содержащий переменные и методы, в том числе два конструктора и функции доступа:

```

1) class Dog
2) {
3)     public String Name;
4)     public String Breed;
5)     public int AverageWeight;
6)     public int AverageHeight;
7)     public void ShowDog(Graphics g, int x, int y)
8)     {
9)         g.drawString("Имя: "+Name, x,y);
10)        g.drawString("Порода: "+Breed, x,y+20);
11)        if (AverageWeight!=-1)
12)            g.drawString("Вес: "+AverageWeight, x,y+40);
13)        if (AverageHeight!=-1)
14)            g.drawString("Рост: "+AverageHeight, x,y+60);
15)    }
16)    public Dog(String name, String breed, int weight, int height)
17)    {
18)        Name=name;
19)        Breed=breed;
20)        AverageWeight=weight;

```



```

21)         AverageHeight=height;
22)     }
23)     public Dog(String name)
24)     {
25)         Name=name;
26)         Breed="Неизвестна";
27)         AverageWeight=-1;
28)         AverageHeight=-1;
29)     }
30)     private String Name;
31)     private String Breed;
32)     private int AverageWeight;
33)     private int AverageHeight;
34)
35)     public void setName(String Name)
36)     {     this.Name=Name; }
37)
38)     public String getName()
39)     {     return Name;     }
40)
41)     public void setBreed(String Breed)
42)     {     this.Breed=Breed; }
43)
44)     public String getBreed()
45)     {     return Breed;     }
46)
47)     public void setAverageWeight(int AverageWeight)
48)     {     this.AverageWeight=AverageWeight; }
49)
50)     public int getAverageWeight()
51)     {     return AverageWeight; }
52)
53)     public void setAverageHeight(int AverageHeight)
54)     {     this.AverageHeight=AverageHeight; }
55)
56)     public int getAverageHeight()
57)     {     return AverageHeight; }
58)
59) }

```

2.2 Индивидуальные задания

Общая часть.

Для своего варианта (см. ниже) выполнить следующее:

- 1) Определить пользовательский класс в соответствии с вариантом задания.
- 2) Определить статический элемент – счетчик.
- 3) Определить в классе следующие конструкторы: без параметров, с параметрами. Каждый конструктор должен выводить сообщение о количестве объектов.
- 4) Определить в классе внешние компоненты-функции для получения и установки полей данных.
- 5) Написать демонстрационную программу, в которой объекты пользовательского класса создаются с помощью неявного использования конструкторов без параметра.
- 6) Показать в программе явное использование конструкторов с параметрами.

2.3 Варианты задания

- 1) Товар
- 2) Тракторист
- 3) Бригада
- 4) Бригадир
- 5) Сотрудник
- 6) Бухгалтер
- 7) Организация
- 8) Ребенок
- 9) Школьник
- 10) Школа
- 11) Выпускник ВУЗа
- 12) Дипломат
- 13) Член партии
- 14) Государство
- 15) Поликлиника
- 16) Банк
- 17) Отдел кадров
- 18) Предприятие
- 19) Автобус
- 20) Квартира
- 21) Улица
- 22) Город
- 23) Страна
- 24) Вертолет
- 25) Телефон
- 26) Телевизор
- 27) Рекламный щит
- 28) Учитель

- 29) Житель города
- 30) Отдел
- 31) Сборная страны
- 32) Спортсмен
- 33) Кафедра
- 34) Преподаватель
- 35) Таксист
- 36) Программист
- 37) Собственник жилья
- 38) Призывник
- 39) Выпускник школы
- 40) Военнослужащий
- 41) Офицер
- 42) Продавец
- 43) Магазин
- 44) Дипломированный специалист
- 45) Врач
- 46) Больница
- 47) Гараж
- 48) Лекарство
- 49) Аптека
- 50) Автомобиль
- 51) Автобус
- 52) Автопарк
- 53) Шофер
- 54) Инженер
- 55) Танк
- 56) Самолет
- 57) Корабль
- 58) Музыкант
- 59) Оркестр
- 60) Выпускник ПТУ

Примечание: идентификаторы переменных и функций должны соответствовать заданию. Пример выполнения индивидуального задания для варианта 000.

```

import java.util.*;
public class var0 {
    public static void main(String[] args) {
        Publ y= new Publ("cc", 8, 9);
        y.Show();
        Publ x = new Publ();
        Publ x1 = new Publ();
        Publ x2 = new Publ();
        y.Read();
        y.Show();
    }
}
class Publ {
    static int count =0;
    String name;
    int numPages, year;
    public Publ(String nm, int np, int y){
        name=nm; numPages=np; year=y;count++;
        System.out.println(count);
    }
    public Publ(){
        count++;
        System.out.println(count);
    }
    public void Read(){
        Scanner in = new Scanner(System.in);
        try {
            System.out.println("vvedite name");
            name=in.nextLine();
            System.out.println("vvedite numPages");
            numPages=in.nextInt();
            System.out.println("vvedite year");
            year=in.nextInt();
        } catch (Exception e) {
            System.out.println("input error");
        }
    }
    public void Show(){
        System.out.println("name="+name+"  numPages="+numPages+"  year= "+year)
    }
    void setName(String n){
        name=n;
    }
    String getName(){
        return name;
    }
    void setNumPages(int n){
        numPages=n;
    }
    int getNumPages(){
        return numPages;
    }
    void setYear(int n){
        year=n;
    }
    int getYear(){
        return year;
    }
}

```

Рисунок 2.1- Текст программы

2.4 Постановка задачи

2.4.1 Определить пользовательский класс «Товар».

2.4.2 Определить статический элемент – счетчик.

2.4.3 Определить в классе следующие конструкторы: без параметров, с параметрами. Каждый конструктор должен выводить сообщение о количестве объектов.

2.4.4 Определить в классе внешние компоненты-функции для получения и установки полей данных.

2.4.5 Написать демонстрационную программу, в которой объекты пользовательского класса создаются с помощью неявного использования конструкторов без параметра.

2.4.6 Показать в программе явное использование конструкторов с параметрами.

```
Microsoft Windows XP [Версия 5.1.2600]
(C) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\ep>d:
D:\>cd java/lab2/var0/bin
D:\Java\lab2\var0\bin>java var0
1
name=cc  numPages=8  year= 9
2
3
4
vvedite name
hkds
vvedite numPages
klds
input error
name=hkds  numPages=8  year= 9
D:\Java\lab2\var0\bin>java var0
1
name=cc  numPages=8  year= 9
2
3
4
vvedite name
khkjhks
vvedite numPages
98908
vvedite year
98789
name=khkjhks  numPages=98908  year= 98789
D:\Java\lab2\var0\bin>
```

Рисунок 2.2

Определение пользовательского класса.

Класс «Товар»

Поля:

- 1) Название – массив символов.
- 2) Цена за единицу – вещественное число.

Пример: Сахар – 120 тенге за кг.

2.5 Дополнительные вопросы и задания

2.5.1 Запустить приложение в IDE

2.5.2 Запустить приложение в командной строке

2.5.3 Скомпилировать файл из командной строки

2.5.4 Прокомментировать произвольное место в коде

2.5.5 Инкапсуляция

2.5.6 Конструкторы

2.5.7 Модификаторы доступа

2.6.8 Статические члены

3 Лабораторная работа №3. Java, наследование

Цель работы:

- ознакомиться с использованием принципа наследования в Java;
- пакеты.

3.1 Порядок выполнения работы

С помощью механизма наследования, на основе класса «Квадрат», создать классы «Прямоугольник», «Равносторонний треугольник». Из класса «Равносторонний треугольник» создать класс «Равнобедренный треугольник». В каждом из указанных классов реализовать внешний компонент «Вычисление площади». Написать демонстрационную программу, иллюстрирующую работу с объектами созданных классов.

Переписать нижеследующий код C++ на Java

- 1) `#include<iostream>`
- 2) `#include<cmath>`
- 3) `using namespace std;`
- 4) `// класс Квадрат - описание`
- 5) `class Tsquare`
- 6) `{`
- 7) `float a;`
- 8) `public:`

```

9)     Tsquare(float);
10)    void set_a(float);
11)    float get_a();
12)    float area();
13)    };
14)    // класс Квадрат - реализация
15)    Tsquare::Tsquare(float a1) {a=a1;};
16)    void Tsquare::set_a(float a1) {a=a1;};
17)    float Tsquare::get_a() {return a;};
18)    float Tsquare::area() {return a*a;};
19)
20)    // класс Прямоугольник - описание
21)    class Trectangle: public Tsquare
22)    {
23)        float b;
24)    public:
25)        Trectangle(float, float);
26)        float area();
27)        };
28)    // класс Прямоугольник - реализация
29)    Trectangle::Trectangle(float a1, float b1): Tsquare(a1) {b=b1;};
30)    float Trectangle::area(){return get_a()*b;};
31)
32)
33)
34)
35)    // класс Равносторонний треугольник - описание
36)    class TequilateralTriag: public Tsquare
37)    {
38)    public:
39)        TequilateralTriag(float);
40)        float area();
41)        };
42)    // класс Равносторонний треугольник - реализация
43)    TequilateralTriag::TequilateralTriag(float a1):Tsquare(a1){};
44)    float TequilateralTriag::area(){return sqrt(3.)/4.*get_a()*get_a();};
45)
46)    // класс Равнобедренный треугольник - описание

```

```

47) class TisosclsTriag: public Tsquare
48) {
49)     float angle_degree;
50) public:
51)     TisosclsTriag(float, float);
52)     float area();
53)     };
54) // класс Равнобедренный треугольник - реализация
55) TisosclsTriag::TisosclsTriag(float a1, float
angle_degree1):Tsquare(a1){angle_degree=angle_degree1;};
56) float TisosclsTriag::area(){float angle_rad= angle_degree*3.14/180.;
return get_a()*get_a()*sin(angle_rad)*0.5;};
57)
58) main()
59)
60) {
61)     Tsquare d(3); cout<<d.area()<<endl;
62)     Trectangle r(2,4); cout<<r.area()<<endl;
63)     TequilateralTriag a(4); cout<<a.area()<<endl;
64)     TisosclsTriag u(5, 90.); cout<<u.area()<<endl;
65)     system("pause");
66) }

```

```

9
8
6.9282
12.5
Для продолжения нажмите любую клавишу . . .

```

Рисунок 3.1- Вид экрана

3.2 Индивидуальные задания

Общая часть.

Для своего варианта выполнить следующее:

1) Построить модель предметной области в соответствии со своим вариантом (см. ниже).

2) Для каждого класса создать конструктор и деструктор, выдающий сообщение о своей работе.

3) Для каждого класса создать внешние функции установки и получения полей данных.

4) Для каждого класса разработать функции, позволяющие представить на экране значения полей данных.

5) Для каждого класса разработать функции, позволяющие вводить с консоли значения полей данных.

6) Написать демонстрационную программу, иллюстрирующую поочередный вызов конструкторов и деструкторов базового и производного классов.

7) Построить диаграмму классов.

3.3 Варианты заданий

- 1) Учебник– Книга - Печатное издание
- 2) Преподаватель-Руководитель дипломной работы – Сотрудник
Университета
- 3) Организация –Магазин -Склад
- 4) Житель города -Работник-Школьник
- 5) Школьник – Сотрудник - Персона
- 6) Студент– Спортсмен - Персона
- 7) Педиатр - Врач- Выпускник медицинского вуза
- 8) Школьник – Персона – Учитель
- 9) Животное – Домашнее животное – Собака
- 10) Персона – Военнообязанный - Призывник
- 11) Спортсмен-Персона-Призер
- 12) Музыкант-Сотрудник-Скрипач
- 13) Фильм-Телефильм-Мультфильм
- 14) Актер-Сотрудник-Режиссер
- 15) Студент-Учащийся-Магистрант
- 16) Военнослужащий-Офицер-Человек
- 17) Человек-Продавец-Шофер
- 18) Дипломированный специалист-Врач-Терапевт
- 19) Медик -Врач-Стоматолог
- 20) Постройка –Гараж -Сарай
- 21) Студент – Преподаватель - Персона
- 22) Врач – Пациент - Персона
- 23) Учитель – Школьник - Персона
- 24) Арендатор – Собственник жилья – Персона
- 25) Офицер – Призывник-Выпускник школы
- 26) Ребенок – Налогоплательщик - Персона
- 27) Организация-Библиотека-Школа
- 28) Автомобиль-Автобус-Такси
- 29) Шофер-Таксист-Водитель автобуса
- 30) Инженер-Программист-Менеджер

- 31) Механизм-Танк-Самолет
 - 32) Военнослужащий-Снайпер-Офицер
 - 33) Плавающее средство –Корабль -Моторная лодка
 - 34) Выпускник консерватории - Музыкант-Персона
 - 35) Выпускник ПТУ-Тракторист-Рабочий
 - 36) Тракторист-Бригадир-Персона
 - 37) Сотрудник-Бухгалтер-Инженер
 - 38) Выпускник АУЭС- Сотрудник- Выпускник вуза
 - 39) Ребенок-Школьник-Спортсмен
 - 40) Выпускник ВУЗа- Дипломат - Инженер
 - 41) Гражданин-Член партии -Персона
 - 42) Государство-Член Совета безопасности – Член НАТО
 - 43) Картина - Рисунок-Экспонат
 - 44) Академик-Ученый-Выпускник вуза
 - 45) Руководитель дипломной работы -Преподаватель-Сотрудник
 - 46) Животное – Домашнее животное – Собака
 - 47) Персона – Военнообязанный - Призывник
 - 48) Спортсмен-Персона-Призер
 - 49) Музыкант-Сотрудник-Скрипач
 - 50) Фильм-Телефильм-Мультфильм
 - 51) Актер-Сотрудник-Режиссер
 - 52) Студент-Учащийся-Магистрант
 - 53) Военнослужащий-Офицер-Человек
 - 54) Человек-Продавец-Шофер
 - 55) Дипломированный специалист-Врач-Терапевт
 - 56) Медик -Врач-Стоматолог
 - 57) Постройка –Гараж -Сарай
 - 58) Лечебное средство-Лекарство-Настойка
 - 59) Студент-Персона-Дипломник
 - 60) Медицинский работник-Врач-Санитарка
- Пример выполнения индивидуального задания для варианта 000

Постановка задачи.

Для варианта «Учебник– Книга - Печатное издание» выполнить следующее:

- 1) Построить модель предметной области в соответствии со своим вариантом (см. ниже).
- 2) Разработать классы в соответствии с моделью предметной области.
- 3) Для каждого класса создать конструктор.
- 4) Для каждого класса создать внешние функции установки и получения полей данных.
- 5) Для каждого класса разработать функции, позволяющие представить на экране значения полей данных.

6) Для каждого класса разработать функции, позволяющие вводить с консоли значения полей данных.

7) Написать демонстрационную программу, иллюстрирующую использование разработанных классов.

8) Построить диаграмму классов.

9) Разработанные классы разместить в одном пакете.

Модель предметной области.

Печатное издание имеет название, число страниц, год издания. Название печатного издания – текстовое поле (состоит из цифр, букв кириллицы и латиницы, минимальное число символов – 3, максимальное число символов – 70). Число страниц – положительное число с максимальным значением меньше 10000. Год издания – четырехзначное положительное число, больше (равно) 1900 и меньше (равно) текущего года.

Альтернативные варианты классов, производных из печатных изданий: Книга, Журнал, Газета, Рекламный буклет, Статья, ...

Каждая Книга является Печатным изданием и следовательно имеет название, число страниц, год издания. Кроме того, каждая книга имеет автора и название издательства. Автор – текстовое поле (состоит только из букв кириллицы, точек и запятых, первая буква – заглавная, минимальное число символов – 3, максимальное число символов – 30). Название издательства – текстовое поле (состоит из букв кириллицы и латиницы, минимальное число символов – 3, максимальное число символов – 15).

Описание альтернативных вариантов классов, производных из печатного издания.

Журнал характеризуется месяцем издания, адресом, типографией. Газета характеризуется месяцем и днем выпуска. Статья характеризуется автором, числом страниц. Рекламный буклет характеризуется организацией, видом услуг. Статья может входить в книгу.

Все текстовые поля могут содержать пробелы.

Каждый учебник является книгой и следовательно имеет название, число страниц, год издания, автора и название издательства. Кроме того, каждый учебник имеет название предмета и номер класса.

Таблица 1 - Примеры объектов

| Название | Числ. стр. | Год изд. | Автор | Назв.изд. | Назв.пр едм. | Номер класса |
|-------------------------|------------|----------|------------|--------------------|--------------|--------------|
| Учебник | | | | | | |
| Сборник задач по физике | 345 | 2012 | Кирик | Мектеп | Физика | 8 |
| Физика | 457 | 2013 | Балгимбаев | Атамура | Физика | 8 |
| Книга | | | | | | |
| Дядя Степа | 25 | 1978 | Михалков | Детская литература | | |

| Публикация | | | | | | |
|--|-----|------|--|--|--|--|
| Сборник трудов научной конференции «Актуальные проблемы энергетики » | 687 | 2013 | | | | |

```

import ErLib.*;

public class inheritance {

    /**
     * @param args
     */

    public static void main(String[] args) {

        Publ f = new Publ();
        f.setNumPages(78); f.Show();
        Book x = new Book();
        x.setName("kjdkljkjkl"); x.Show();
        ClassBook d = new ClassBook("hhh", 45, 78, "jj", "lliu", "rrt", 72);
        d.Show();

    }
}

```

Рисунок 3.2-Текст программы

```

D:\Java\lab3\inheritance\bin>java inheritance
1
null 78 02
kjdkljkjkl 0 0 null null3
hhh 45 78 jj lliu rrt 72
D:\Java\lab3\inheritance\bin>

```

Рисунок 3.3

```

package ErLib;
import java.util.*;
public class Publ {

    static int count;
    String name;
    int numPages, year;
    public Publ(String nm, int np, int y){
        name=nm; numPages=np; year=y;count++;
        System.out.println(count);
    }
    public Publ(){
        count++;
        System.out.println(count);
    }
    public void Read(){
        Scanner in = new Scanner(System.in);
        System.out.println("Введите name");
        name=in.nextLine();
        System.out.println("Введите numPages");
        numPages=in.nextInt();
        System.out.println("Введите year");
        year=in.nextInt();
    }
    public void Show(){
        System.out.print(name+" "+numPages+" "+year);
    }
    public void setName(String n){
        name=n;
    }
    public String getName(){
        return name;
    }
    public void setNumPages(int n){
        numPages=n;
    }
    public int getNumPages(){
        return numPages;
    }
    public void setYear(int n){
        year=n;
    }
    public int getYear(){
        return year;
    }
}

```

Рисунок 3.4 -Текст программы

```

package ErLib;

import java.util.Scanner;

public class Book extends Publ{
    String author, publishingHouse;
    public Book(){

    }
    public Book(String nm, int np, int y, String at,
                String ph){
        super(nm, np, y);
        author=at; publishingHouse=ph;
    }
    public void setAuthor(String a){
        author=a;
    }
    public String getAuthor(){
        return author;
    }
    public void setPublishingHouse(String a){
        publishingHouse=a;
    }
    public String getPublishingHouse(){
        return publishingHouse;
    }
    public void Show(){
        super.Show();
        System.out.print("  "+author+"  "+publishingHouse);
    }
    public void Read(){
        super.Read();
        Scanner in = new Scanner(System.in);
        System.out.println("Введите author");
        author=in.nextLine();
        System.out.println("Введите PublishingHouse");
        numPages=in.nextInt();
    }
}







```

Рисунок 3.5 -Текст программы

[ClassBook.java]-BRED2

Фл Правка Выравнивание Разное Поиск

```







/**
 * @param n
 * @roseuid 520E4564034B
 */
public void setSubject (String n)
{
    subject=n;
}

/**
 * @return java.lang.String
 * @roseuid 520E4564034D
 */
public String getName ()
{
    return subject;
}

/**
 * @param n
 * @roseuid 520E4564035B
 */
public void setNClass (int n)
{
    nClass=n;
}

/**
 * @return int
 * @roseuid 520E4564035D
 */
public int getNumPages ()
{
    return numPages;
}

```

Рисунок 3.6 -Текст программы

3.4 Дополнительные вопросы и задания

- 3.4.1 Запустить приложение в IDE.
- 3.4.2 Запустить приложение в командной строке.
- 3.4.3 Скомпилировать файл из командной строки.
- 3.4.4 Прокомментировать произвольное место в коде.
- 3.4.5 Наследование.
- 3.4.6 Пакеты.
- 3.4.7 Инкапсуляция.
- 3.4.8 Конструкторы.
- 3.4.9 Модификаторы доступа.
- 3.4.10 Статические члены.

4 Лабораторная работа №4. Java, массивы, интерфейс comparable

Цель работы:

- ознакомиться с использованием массивов в Java;
- научиться сортировать массивы объектов на основе интерфейса Comparable;
- разобраться с проблемой копирования объектов;
- получить представление об обратном проектировании с использованием доступных case средств.

4.1 Порядок выполнения работы

Описать класс «Треугольник» (три стороны). Разработать: функцию для определения площади треугольника, конструктор с параметрами, перезагрузить оператор вывода. Создать массив объектов класса «Треугольник». Написать программу, иллюстрирующую приемы работы с массивом объектов.

Текст программы (переписать на Java)

```
#include<iostream>  
#include<cmath>  
using namespace std;  
  
// класс Треугольник - описание  
class Ttriangle  
{  
    float a, b, c;  
public:
```



```

    Ttriangle(float, float, float);
    Ttriangle(){};
    float area();
    friend ostream & operator<<(ostream & os, const Ttriangle & t);
};
// класс Треугольник - реализация
Ttriangle::Ttriangle(float a1, float b1, float c1) {a=a1;b=b1; c=c1;};
float Ttriangle::area() {float p=(a+b+c)/2.; return sqrt(p*(p-a)*(p-b)*(p-c));};
// Формула Герона
ostream & operator<<(ostream & os, const Ttriangle & t)
    {os<<"Ttriangle: "<< t.a << " "<<t.b<<" "<<t.c<<endl;};

main()
{
    Ttriangle b[1000] = {Ttriangle(2, 3, 4), Ttriangle(5, 3, 4), Ttriangle(5, 3, 4),
    Ttriangle(7, 3, 4) };
    int i; for (i=0; i<=3; i++) {cout<<b[i]<<endl; };
    cout<<sizeof(Ttriangle)<<endl;
    cout<<sizeof(b)<<endl;
    system("pause");
}

```

```

Ttriangle: 2 3 4
Ttriangle: 5 3 4
Ttriangle: 5 3 4
Ttriangle: 7 3 4
12
12000
Для продолжения нажмите любую клавишу

```

Рисунок 4.1 - Вид экрана

Описать класс «Треугольники» (число треугольников, массив треугольников). Разработать: функцию добавления элемента класса «треугольник», функцию для определения общей площади всех треугольников.

Написать программу, иллюстрирующую приемы работы с классом «Треугольники».

Текст программы (переписать на Java).

```

*/
#include<iostream>
#include<cmath>

```

```

#define max_number_triangles 1000
using namespace std;

// класс Треугольник - описание
class Ttriangle
{
    float a, b, c;
public:
    Ttriangle();
    Ttriangle(float, float, float);
    float area();
    friend ostream & operator<<(ostream & os, const Ttriangle & t);
};
// класс Треугольник - реализация
Ttriangle::Ttriangle(){};
Ttriangle::Ttriangle(float a1, float b1, float c1) {a=a1;b=b1; c=c1;};
float Ttriangle::area() {float p=(a+b+c)/2.; return sqrt(p*(p-a)*(p-b)*(p-c));};
// Формула Герона
ostream & operator<<(ostream & os, const Ttriangle & t)
    {os<<"Ttriangle: "<< t.a << " "<<t.b<< " "<<t.c<<endl;};

// класс Треугольники - описание
class Ttriangles
{
    int number_triangles;
    Ttriangle triangles[max_number_triangles];
public:
    Ttriangles();
    void add_triangle(float, float, float);
    float area();
    friend ostream & operator<<(ostream & os, const Ttriangles & t);
};
// класс Треугольники - реализация
Ttriangles::Ttriangles(){number_triangles=0;};
void Ttriangles::add_triangle(float a , float b, float c)
{number_triangles=number_triangles+1;
  Ttriangle t(a, b, c);
  triangles[number_triangles-1]=t;
};
float Ttriangles::area()
{float s=0; int i;
  for (i=1; i<=number_triangles; i++)
    {s=s+triangles[i-1].area();
    };
};

```

```

    return s;
};
ostream & operator<<(ostream & os, const Ttriangles & t)
{
int i; for (i=1; i<= t.number_triangles; i++)
    { os<<"Ttriangles: "<< t.triangles[i-1]<<endl;
      };
};
main()
{
Ttriangles d;
d.add_triangle(5, 3, 4); d.add_triangle(5, 3, 6); d.add_triangle(5, 4, 4);
d.add_triangle(5, 6, 4);
cout<<d.area()<<endl;
cout<<d;
cout<<sizeof(d)<<endl;

system("pause");
}

```

```

31.2111
Ttriangles: Ttriangle: 5 3 4
Ttriangles: Ttriangle: 5 3 6
Ttriangles: Ttriangle: 5 4 4
Ttriangles: Ttriangle: 5 6 4
12004
Для продолжения нажмите любую клавишу . . .

```

Рисунок 4.2 -Вид экрана

4.2 Индивидуальные задания

Общая часть.

Для своего варианта разработать и описать модель предметной области. Построить иерархию классов. Для каждого класса определить, минимум, два поля различных типов.

Создать класс, включающий в себя массив экземпляров другого класса. Разработать функции, позволяющие: добавить элемент в конец массива объектов; удалить последний элемент; вывести массив на экран; сортировать массив по произвольному полю.

Разобраться с проблемой клонирования (все объекты – указатели).

Указать все отношения. Построить диаграмму классов с помощью любого case-средства, позволяющего строить диаграммы классов по коду на java.

4.3 Варианты заданий

- 1) Библиотека– Книга - Печатное издание
- 2) Самолет-Истребитель-Эскадрилья
- 3) Плавающее средство-Корабль-Флот
- 4) Выпускник консерватории - Музыкант-Оркестр
- 5) Выпускник ПТУ-Тракторист-Бригада
- 6) Тракторист-Бригадир-Совет бригадиров
- 7) Сотрудник-Бухгалтер-Бухгалтерия
- 8) Выпускник АУЭС- Сотрудник- Организация
- 9) Ребенок-Школьник-Школа
- 10) Выпускник вуза- Дипломат - Министерство иностранных дел
- 11) Гражданин-Член партии -Партия
- 12) Государство-Член Совета безопасности – Совет безопасности
- 13) Картина - Выставка-Экспонат
- 14) Академия-Ученый-Выпускник вуза
- 15) Университет-Преподаватель-Сотрудник
- 16) Данные – Информация – Банк данных
- 17) Музей-Экспонат-Предмет
- 18) Корпорация-Фирма-Отдел
- 19) Музыкант-Сотрудник-Оркестр
- 20) Фильм-Киностудия-Мультфильм
- 21) Актер-Сотрудник-Театр
- 22) Магистратура-Учащийся-Магистрант
- 23) Творческий кружок -Дворец школьников-Школьник
- 24) Архитектура-Архитектор-Персона
- 25) Официант-Ресторан-Работник Ресторана
- 26) Профсоюзный комитет-Работник – Член профсоюза
- 27) Постройка –Гараж -Гаражный кооператив
- 28) Лечебное средство-Лекарство-Аптека
- 29) Общественное объединение-Партия-Парламент
- 30) Медицинский работник-Врач-Поликлиника
- 31) Организация –Магазин -Торговая сеть
- 32) Житель города -Работник-Организация
- 33) Отдел – Сотрудник - Персона
- 34) Сборная страны– Спортсмен - Персона
- 35) Поликлиника - Врач- Выпускник медицинского вуза
- 36) Военнослужащий-Офицер-Штаб
- 37) Человек-Продавец-Магазин
- 38) Дипломированный специалист-Врач-Больница
- 39) Отделение-Врач-Поликлиника
- 40) Постройка –Гараж -Гаражный кооператив
- 41) Лечебное средство-Лекарство-Аптека
- 42) Общественное объединение-Партия-Парламент

- 43) Медицинский работник-Врач-Поликлиника
 - 44) Организация –Магазин -Торговая сеть
 - 45) Житель города -Работник-Организация
 - 46) Отдел – Сотрудник - Персона
 - 47) Сборная страны– Спортсмен - Персона
 - 48) Поликлиника - Врач- Выпускник медицинского вуза
 - 49) Кафедра – Преподаватель - Персона
 - 50) Больница – Пациент - Персона
 - 51) Школа – Школьник - Персона
 - 52) Жилищный кооператив – Собственник жилья – Персона
 - 53) Призыв – Призывник-Выпускник школы
 - 54) Государство – Налогоплательщик - Персона
 - 55) Организация-Библиотека-Библиотекарь
 - 56) Автомобиль-Автобус-Автопарк
 - 57) Шофер-Таксист-Таксопарк
 - 58) Инженер-Программист-Проект
 - 59) Механизм-Танк-Полк
 - 60) Солдат-Снайпер-Отделение
- Пример выполнения индивидуального задания для варианта 000

Постановка задачи.

Для своего варианта разработать и описать модель предметной области. Построить иерархию классов. Для каждого класса определить, минимум, два поля различных типов.

Создать класс, включающий в себя массив экземпляров другого класса. Разработать функции, позволяющие: добавить элемент в конец массива объектов; удалить последний элемент; вывести массив на экран; сортировать массив по произвольному полю.

Разобраться с проблемой клонирования (все объекты – указатели).

Указать все отношения. Построить диаграмму классов с помощью любого case-средства, позволяющего строить диаграммы классов по коду на java.

Описание модели предметной области.

Библиотека характеризуется городом, в котором она расположена и своим уникальным названием. Город и название библиотеки – текстовые поля (состоит только из букв кириллицы, первая буква – заглавная, минимальное число символов – 3, максимальное число символов – 15). Библиотека состоит из книг.

Печатное издание имеет название, число страниц, год издания. Название Печатного издания – текстовое поле (состоит из цифр, букв кириллицы и латиницы, минимальное число символов – 3, максимальное число символов – 70). Число страниц – положительное число с максимальным значением

меньше 10000. Год издания – четырехзначное положительное число, больше (равно) 1900 и меньше (равно) текущего года.

Альтернативные варианты классов, производных из печатных изданий: Книга, Журнал, Газета, Рекламный буклет, Статья, ...

Описание альтернативных вариантов классов, производных из печатного издания:

Журнал характеризуется месяцем издания, адресом, типографией. Газета характеризуется месяцем и днем выпуска. Статья характеризуется автором, числом страниц. Рекламный буклет характеризуется, организацией, видом услуг. Статья может входить в книгу.

Все текстовые поля могут содержать пробелы.

Отношения.

1) Отношение наследования.

Печатное издание – книга. Печатное издание – базовый класс, книга – производный класс. Каждая книга – печатное издание. Не каждое печатное издание – книга. Например, – статья, буклет, газета.

2) Отношение агрегации.

Книга – Библиотека. Библиотека состоит из книг. Книги входят в библиотеку. Библиотека – агрегирующий класс. Книга – агрегируемый класс.

```
import java.util.Arrays;
import ErLib.*;
public class Array {

    public static void main(String[] args) {

        Lib f = new Lib();
        Book b = new Book();
        b.setYear(5); b.setName("xbac"); f.Add(b);
        b.setYear(3); b.setName("abc"); f.Add(b);
        b.setYear(9); b.setName("cbc"); f.Add(b);
        b.setYear(1); b.setName("habc"); f.Add(b);
        f.Show();
        f.Sort();
        System.out.println("jj");
        f.Show();
        // TODO Auto-generated method stub

    }
}
```

Рисунок 4.3 -Текст программы

```

package ErLib;
import java.util.*;
import java.util.Arrays;

public class Lib {
int nb;
Book[] bs = new Book[100];
public Lib(){
    nb=0;
}
public void Add(Book b){
bs[nb] = new Book();
bs[nb]=b.copy();    nb++;
}
public void Sort(){
    Arrays.sort(bs, 0, nb);
}
public void Show(){
    for (int i = 0; i < nb; i++) {
        bs[i].Show(); System.out.println("");
    }
}
}
}

```

Рисунок 4.4

```
package ErLib;
import java.util.*;
import java.util.Arrays;

public class Lib {
    int nb;
    Book[] bs = new Book[100];
    public Lib(){
        nb=0;
    }
    public void Add(Book b){
        bs[nb] = new Book();
        bs[nb]=b.copy();    nb++;
    }
    public void Sort(){
        Arrays.sort(bs, 0, nb);
    }
    public void Show(){
        for (int i = 0; i < nb; i++) {
            bs[i].Show(); System.out.println("");
        }
    }
}
```

4.5 сурет -Текст программы


```
package ErLib;
import java.util.*;

public class Book extends Publ implements Comparable{
    String author, publishingHouse;
    public Book(){

    }
    public Book(String nm, int np, int y, String at,
                String ph){
        super(nm, np, y);
        author=at; publishingHouse=ph;
    }
    public int compareTo(Object obj){
        Book tmp = (Book)obj;
        int res=0;
        if(this.name.compareTo(tmp.name)<0)
        // if(this.year< tmp.year)
        {
            res= -1;
        }
        if(this.name.compareTo(tmp.name)>0)
        // if(this.year > tmp.year)
        {
            res = 1;
        }
        return res;
    }
    public Book copy(){
        Book copy = new Book();
        copy = (Book) super.copy();
        copy.author = this.author;
        copy.publishingHouse=this.publishingHouse;
        return copy;
    }
    public void setAuthor(String a){
        author=a;
    }
    public String getAuthor(){
```

Рисунок 4.6 -Текст программы

```

14.09.2013 12:48 <DIR>      ..
16.08.2013 11:23 <DIR>      .metadata
13.08.2013 09:33 <DIR>      Array
                0 файлов      0 байт
                4 папок  122 547 376 128 байт свободно

D:\Java\lab4>cd array/bin

D:\Java\lab4\Array\bin>java array
Exception in thread "main" java.lang.NoClassDefFoundError: array (wrong name: Array)
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(Unknown Source)
    at java.security.SecureClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.access$100(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)

D:\Java\lab4\Array\bin>dir
Том в устройстве D имеет метку DATA-NEW
Серийный номер тома: B762-FCE5

Содержимое папки D:\Java\lab4\Array\bin

14.09.2013 12:48 <DIR>      .
14.09.2013 12:48 <DIR>      ..
14.09.2013 20:49          889 Array.class
14.09.2013 12:48 <DIR>      ErLib
                1 файлов      889 байт
                3 папок  122 547 376 128 байт свободно

D:\Java\lab4\Array\bin>java Array
xbac 0 5 null null
abc 0 3 null null
cbc 0 9 null null
habc 0 1 null null
jj
abc 0 3 null null
cbc 0 9 null null
habc 0 1 null null
xbac 0 5 null null

D:\Java\lab4\Array\bin>

```

Рисунок 4.7 -Текст программы

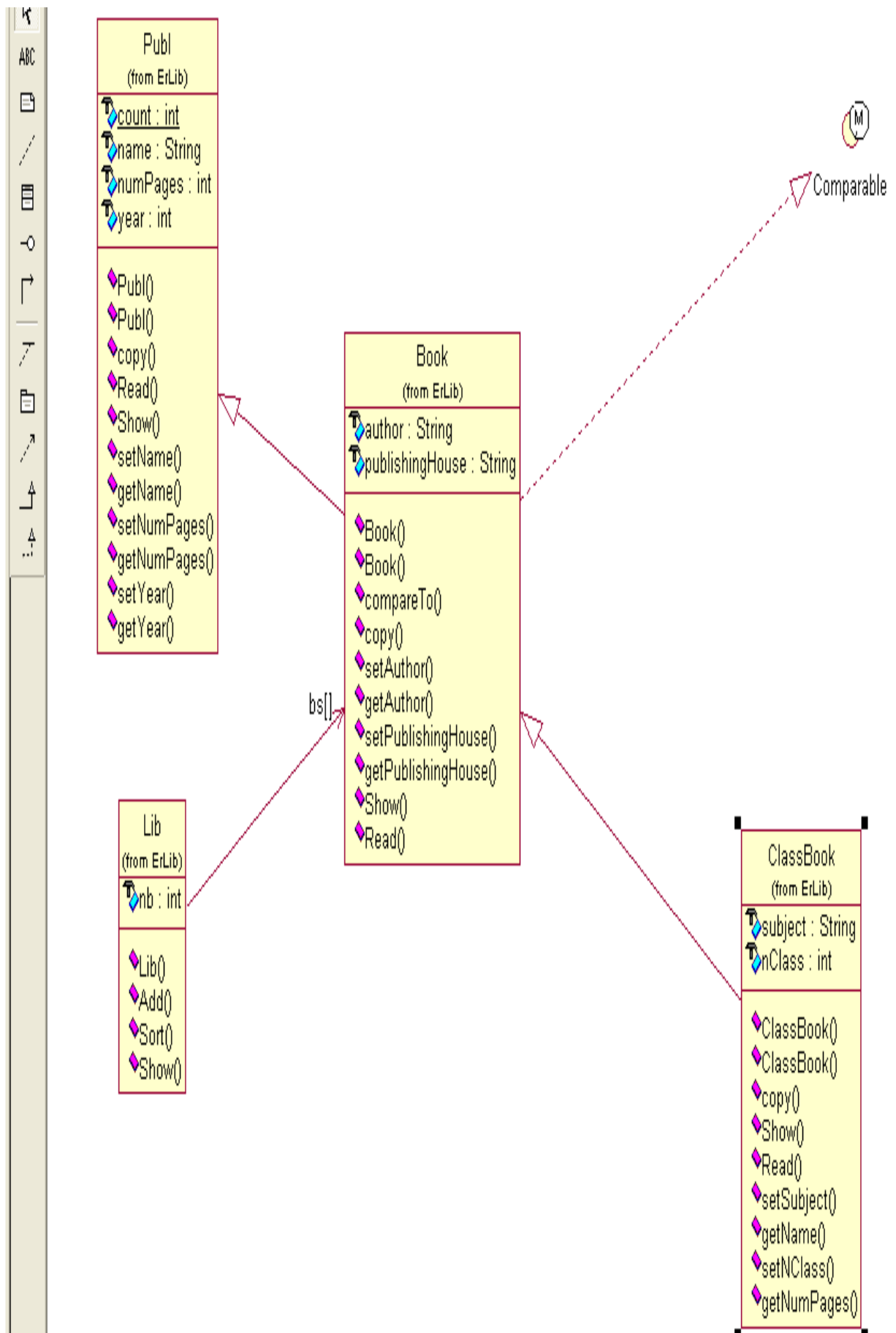


Рисунок 4.8

4.4 Дополнительные вопросы и задания

- 4.4.1 Запустить приложение в IDE.
- 4.4.2 Запустить приложение в командной строке.
- 4.4.3 Прокомментировать произвольное место в коде.
- 4.4.4 Отношения UML и их обозначения на диаграмме классов.
- 4.4.5 Интерфейсы.
- 4.4.6 Интерфейс Comparable.
- 4.4.7 Для чего в варианте 000 была реализована функция COPY.
- 4.4.8 Инструменты обратного проектирования.
- 4.4.9 Наследование.
- 4.4.10 Пакеты.
- 4.4.11 Инкапсуляция.
- 4.4.12 Конструкторы.
- 4.4.13 Модификаторы доступа.
- 4.4.14 Статические члены.

Список литературы

- 1 Шилдт Г. Java. Полное руководство. -М.: Издательский дом «Вильямс», 2012. -1104 с.
- 2 Рябушко А.П. и др. Сборник индивидуальных заданий по высшей математике. Ч.3, 1991. -288 с.
- 3 Основы программирования на С#. -Биллиг В.А. рус. -М., 2006.
- 4 С# начала программирования.- Ишкова Э.А. рус.- М.: «Бином», 2011.
- 5 Полный справочник по С# Шилдт Г. рус. - М.: Издательский дом «Вильямс», 2006.

Турганбаев Ерик Сулейменович
Кожамбердиева Мергул Иманбековна

СОВРЕМЕННЫЕ МЕТОДЫ И СРЕДСТВА ПРОГРАММИРОВАНИЯ

Методические указания к выполнению лабораторных работ
для студентов специальности
5B070400 – Вычислительная техника и программное обеспечение

Редактор Л.М.Сластикина
Специалист по стандартизации Н.К. Молдабекова

Подписано в печать ____ . ____ . ____ .
Тираж __ 50__ экз.
Объем 2,8 уч.- изд. л.

Формат 60x84 1/16
Бумага типографская №1
Заказ _____ Цена 1400 тен.

Копировально-множительное бюро
некоммерческого акционерного общества
«Алматинский университет энергетики и связи»
050013, Алматы, Байтурсынулы, 126