

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 10.11.2023 03:18:59
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
"Юго-Западный государственный университет"
(ЮЗГУ)

Кафедра биомедицинской инженерии



Проректор по учебной работе
О.Г. Локтионова
2017

ПРИКЛАДНАЯ МАТЕМАТИЧЕСКАЯ СТАТИСТИКА

Методические указания к проведению лабораторных работ для
студентов направления подготовки 30.05.03 - "Медицинская
кибернетика"

Курск 2017

УДК 615.478

Составитель Д.Е.Скопин

Рецензент

Доктор технических наук, профессор *И.Е. Чернецкая*

Прикладная математическая статистика: методические указания к проведению лабораторных работ / Юго-Зап. гос. ун-т; сост. Д.Е.Скопин. - Курск, 2017. - 20 с.: ил. 4, табл.2. - Библиогр.: с. 20.

Содержатся сведения, необходимые для выполнения лабораторных работ по прикладной математической статистике.

Методические указания по структуре, содержанию и стилю изложения материала соответствуют методическим и научным требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для студентов направления подготовки 30.05.03 очной и заочной форм обучений.

Текст печатается в авторской редакции

Подписано в печать

Формат 60x84 1/16

Усл. печ.л. . Уч. -изд.л. Тираж 100 экз. Заказ. Бесплатно.

Юго-Западный государственный университет

305040, г.Курск, ул. 50 лет Октября, 94

Лабораторная работа №1

Пакет Microsoft Excel, работа с разнотипными данными, ведение учета, баз данных

1.1. Цель работы: Получить практические навыки при работе с пакетом прикладных программ Microsoft Excel.

1.2. Краткие теоретические сведения. Пакет прикладных программ Microsoft Excel позволяет автоматизировать медико-биологические исследования, а также проводить визуализацию данных, например строить графики функций, в том числе функции от нескольких переменных.

Пример построения графика функции

Задание. Построить график функции, которая задается уравнением в полярных координатах: $\rho = a \cdot \sin(k \cdot \varphi)$, где $a = 2$, $k = 3$.

Рекомендации. Разместить в ячейке **C2** значение коэффициента a (равного 2), в ячейке **I2** – значение коэффициента k (равного 3). Подготовить диапазон изменения координаты φ . Для этого в ячейку **A2** записать 0, в ячейку **A3** записать 0,05 (в числовом формате). Затем выделить содержимое двух ячеек и произвести растаскивания (см выше), например, до ячейки с адресом **A150**.

1. Рассчитать значение функции на данном диапазоне в полярных координатах $\rho = \rho(\varphi)$. Для этого в ячейку **B2** записать формулу: $=2*\text{SIN}(3*A2)$. Вычислить и провести «растяжку» до **B150**.

2. Рассчитать значения x (столбец **D**) и y (столбец **E**) в декартовой системе координат по формулам: $x = \rho \cos \varphi$, $y = \rho \sin \varphi$. Для этого в ячейку **D2** ввести формулу: $=B2*\text{COS}(A2)$, а в ячейку **E2** – формулу: $=B2*\text{SIN}(A2)$. Вычислить и «растянуть» результаты аналогично п. 1.

3. Выделить диапазон области определения и области значения функции (необходимые диапазоны столбцов **D** и **E**, то есть от **D2:E2** до **D150:E150**) и воспользоваться мастером построения диаграмм (**График – Точечная**). Как и в предыдущем примере в **Мастере диаграмм** можно нажимать на кнопку **Далее** без каких либо дополнительных установок (рис. 31).

1.3. Задание для лабораторной работы:

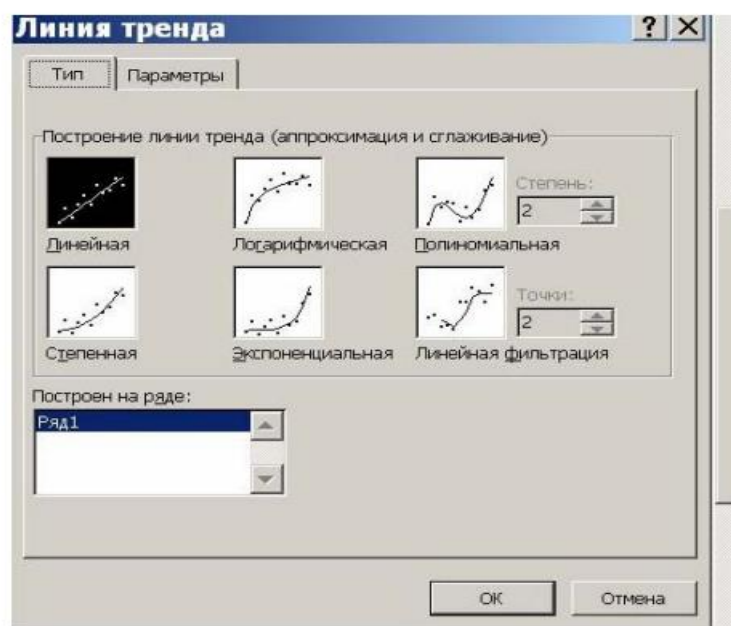
1. Используя пример, описанный выше, получите у преподавателя математическое представление функции и постройте ее график

2. Требуется предсказать объем авиаперевозок на 1961 год. Знание этого объема позволяет планировать развитие авиационной промышленности и инфраструктуры, связанной с авиаперевозками. Исходные данные приведены в таблице.

	А	В
1	Год	Тыс. за год
2	1949	1520
3	1950	1676
4	1951	2042
5	1952	2364
6	1953	2700
7	1954	2867
8	1955	3408
9	1956	3939
10	1957	4421
11	1958	4572
12	1959	5140
13	1960	5714

Порядок расчетов следующий.

1. Выделить диапазон В2:В13 и построить по этим данным диаграмму типа "График", щелкнув по значку "Мастер диаграмм" на панели инструментов.
2. Выделить диаграмму и выполнить Диаграмма/Добавить линию тренда.



3. В окне "Линия тренда" открыть вкладку "Параметры" и установить флажки "Показывать уравнение на диаграмме" и "Поместить на диаграмму величину достоверности аппроксимации".

Постройте графическое изображение диаграммы и получите линейную аппроксимацию данных. Помните, что в качестве аргумента x в уравнении используется номер строки (2-13). Используя уравнение

аппроксимации предскажите, какой пассажиропоток ожидается в 1961 году

3. Создать 2 Java-программы в соответствии с описаниями:

1. Создайте программу, которая сначала попросит пользователя ввести первый номер, второе число и математическую операцию (+, -, /, *), а затем вычислить результат в соответствии с операцией, введенной пользователем:

Enter first number: 5

Enter second number: 6

Enter operation: *

Result: 30

2. Создайте программу, которая позволяет вводить строку, затем вычисляет:

A) Число цифр в строке

B) Количество заглавных букв

C) Количество пробелов в строке

Например:

Enter the string: Hello 1 world 2 My Name is 5 Vasya

Result: 3 digits 4 capital chars 8 spaces

1.4 Содержание отчета

1. Титульный лист

2. Задание на лабораторную работу

3. Листинг программ, графики функций, предсказанные значения

4. Скриншоты работы приложений

Лабораторная работа №2.

Пакет Microsoft Excel, статистические и математические функции

2.1 Цель работы: Изучение фундаментальных блоков программы Microsoft Excel, программирование макросов, с использованием консоли операционной системы Windows, использование пакета для проведения статистических и математических исследований.

2.2. Краткие теоретические сведения

Java - это язык программирования, первоначально разработанный Джеймсом Гослингом в Sun Microsystems (который с тех пор объединился с Oracle Corporation), и выпущен в 1995 году как основной компонент платформы Java Sun Microsystems. Язык унаследовал большую часть своего синтаксиса из C и C++, но имеет более простую объектную модель и возможности более низкого уровня. Приложения Java обычно скомпилированы в байт-код (файл класса), который может работать на любой виртуальной машине Java (JVM) независимо от архитектуры компьютера. Java - это универсальный, параллельный, основанный на классе, объектно-ориентированный язык, который специально разработан для того, чтобы иметь как можно меньше зависимостей в реализации. Он предназначен для того, чтобы разработчики приложений «написал один раз, запустил в любом месте», что означает, что код, который работает на одной платформе, не нужно перекомпилировать для запуска на другой. В настоящее время Java является одним из самых популярных языков программирования, особенно для клиент-серверных веб-приложений, с сообщением о 10 миллионах пользователей.

Разработчики Java решили использовать комбинацию компиляции и интерпретации. Программы, написанные на Java, скомпилированы в машинный язык, но этот язык для компьютера, который на самом деле не существует. Этот так называемый «виртуальный» компьютер известен как виртуальная машина Java или JVM. Язык для виртуальной машины Java называется байт-кодом Java. Нет причин, по которым байт-код нельзя использовать в качестве машинного языка реального компьютера, а не виртуального. Но на самом деле использование виртуальной машины делает возможным одну из основных точек продажи Java: тот факт, что ее можно фактически использовать на любом компьютере. Все, что требуется компьютеру, - это интерпретатор для байт-кода Java. Такой интерпретатор имитирует JVM так же, как Virtual PC имитирует компьютер ПК. (Термин JVM также используется для программы интерпретатора байт-кода Java, которая выполняет симуляцию, поэтому мы говорим, что компьютеру требуется JVM для запуска программ Java. Технически было бы правильнее сказать, что интерпретатор реализует JVM, чем сказать, что это JVM.)

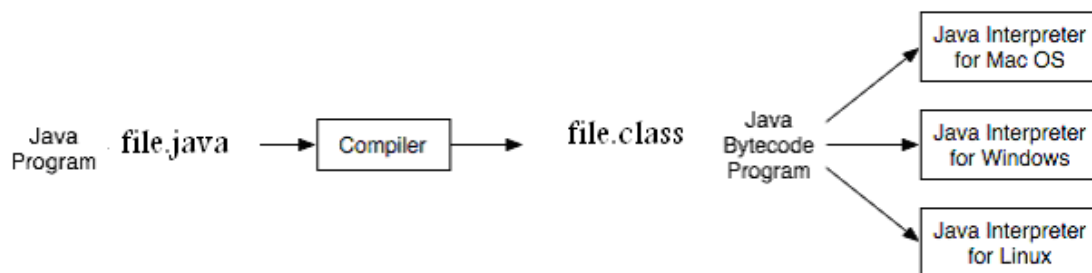


Рисунок 2.1. Создание и исполнение байт-кода

Рисунок 2.1. Иллюстрирует основные этапы создания и выполнения байтового кода. Сначала программист создает оригинальный файл Java, который должен быть сохранен в файле с расширением «.java», это текстовый файл, который можно создать с помощью любого текстового редактора. Затем исходный файл .java может быть скомпилирован в байт-код с использованием компилятора Java. В операционной системе Windows можно использовать консольную команду:

«Javac FileName.java», где «FileName.java» - это фактическое имя файла Java, созданного программистом. Приведенный байт-код после компиляции будет сохранен под тем же именем с расширением .class. Затем байт-код можно выполнить с помощью виртуальной машины Java командой: "java имя_файла".

Как правило, для создания программы, написанной на VB.Net, C ++, C #, вам нужен как минимум компилятор, интегрированный в IDE, такой как Microsoft Visual Studio 2008, 2010. Чтобы создать программу на Java, вы также можете выбрать подходящие IDE, такие как Eclipse или NetBeans или, если хотите, вы можете создавать и компилировать программу Java без какой-либо IDE, используя консольный режим операционной системы. Сегодня мы покажем, как создать простую программу Java, используя консольный режим, затем с помощью Eclipse и, наконец, с помощью IDE NetBeans.

2.3. Порядок выполнения работы

1. Внесите исходные данные
2. По каждому столбцу рассчитайте статистические функции, заданные преподавателем

А	Б	В	Г	Д
62	69	68	38	263
63	119	70	77	229
47	43	55	55	278
42	97	60	77	237

55	89	68	59	242
25	72	33	82	255
70	72	80	76	261
33	115	41	93	236
50	68	51	82	260
52	115	62	49	226
59	86	71	77	252
54	-1	71	78	312
33	24	50	82	294
92	77	95	40	256
86	67	98	5	269
31	42	41	61	285
31	110	35	60	239
76	39	88	34	283
84	78	96	86	262
27	90	32	95	250
49	71	51	49	257

3. Постройте корреляционную матрицу столбцов А-Д
4. Выберите из корреляционной матрицы такие пары столбцов, которые вы считаете взаимозависимыми.
5. Постройте точечные графики взаимозависимых столбцов и оцените вид зависимости.
6. Выберите 2 столбца, которые имеют самые низкие коэффициенты взаимной корреляции
7. Используя т-тест Стьюдента оцените, являются ли данные этих столбцов статистически различными с вероятностью ошибки меньше 0.05

2.4. Содержание отчета

1. Задание на лабораторную работу
2. Последовательность действий, произведенных при выполнении задания
3. Текст программы
4. Скриншот работы созданного приложения

Лабораторная работа №3

Пакет Matlab, введение в систему обработки данных

3.1. Цель работы: изучить основы работы со средой Matlab, а также использование данной среды для проведения статистических и математических расчетов

3.2. Краткие теоретические сведения

Система MATLAB создана таким образом, что любые (подчас весьма сложные) вычисления можно выполнять в режиме прямых вычислений, то есть без подготовки программы. Это превращает MATLAB в необычайно мощный калькулятор, который способен производить не только обычные для калькуляторов вычисления (например, выполнять арифметические операции и вычислять элементарные функции), но и операции с векторами и матрицами, комплексными числами, рядами и полиномами. Можно почти мгновенно задать и вывести графики различных функций – от простой синусоиды до сложной трехмерной фигуры.

Работа с системой в режиме прямых вычислений (или в командном режиме) носит диалоговый характер и происходит по правилу «задал вопрос, получил ответ». Пользователь набирает на клавиатуре вычисляемое выражение, редактирует его (если нужно) в командной строке и завершает ввод нажатием клавиши ENTER.

Даже из таких простых примеров можно сделать некоторые выводы:

- для указания ввода исходных данных используется символ >>;
- данные вводятся с помощью простейшего строчного редактора;
- для блокировки вывода результата вычислений некоторого выражения после него надо установить знак «;» (точка с запятой);
- если не указана переменная для значения результата вычислений, то MATLAB назначает такую переменную с именем ans;

• знаком присваивания является привычный математикам знак равенства =, а не комбинированный знак :=, как во многих других языках программирования и математических системах;

• встроенные функции (например sin) записываются строчными буквами, и их аргументы указываются в круглых скобках;

О переносе строки в сессии

В некоторых случаях вводимое математическое выражение может оказаться настолько длинным, что для него не хватит одной строки. В этом случае часть выражения можно перенести на новую строку с помощью знака многоточия «...» (3 или более точек), например:

$s = 1 - 1/2 + 1/3 - 1/4 + 1/5 - 1/6 + 1/7 - \dots$

$1/8 + 1/9 - 1/10 + 1/11 - 1/12;$

Этот прием может быть весьма полезным для создания наглядных документов, у которых предотвращается заход строк в невидимую область окна.

Центральным понятием всех математических систем является математическое выражение. Оно задает то, что должно быть вычислено в численном (реже символьном) виде. Вот примеры простых математических выражений:

$2+3$

$2.301*\sin(x) 4+\exp(3)/5 \sqrt{y}/2 \sin(\pi/2)$

Математические выражения строятся на основе чисел, констант, переменных, операторов, функций и разных спецзнаков. Ниже даются краткие пояснения сути этих понятий.

Число — простейший объект языка MATLAB, представляющий количественные данные. Числа можно считать константами, имена которых совпадают с их значениями. Числа используются в общепринятом представлении о них. Они могут быть целыми, дробными, с фиксированной и плавающей точкой. Возможно представление чисел в хорошо известном научном формате с указанием мантиссы и порядка числа. Ниже приводятся примеры представления чисел:

0

2 -3

2.301

0.00001

123.456e-24-234.456e10

Как нетрудно заметить, в мантиссе чисел целая часть отделяется от дробной не запятой, а точкой, как принято в большинстве языков программирования. Для отделения порядка числа от мантиссы используется символ e . Знак «плюс» у чисел не проставляется, а знак «минус» у числа называют унарным минусом. Пробелы между символами в числах не допускаются. Числа могут быть комплексными: $z = \text{Re}(z) + \text{Im}(z)*i$. Такие числа содержат действительную $\text{Re}(z)$ и мнимую $\text{Im}(z)$ части. Мнимая часть имеет множитель i или j , означающий корень квадратный из -1:

$3i$

$2+3i -123.456+2.7e-3i$

Функция $\text{real}(z)$ возвращает действительную часть комплексного числа, $\text{Re}(z)$, а функция $\text{imag}(z)$ – мнимую, $\text{Im}(z)$. Для получения модуля комплексного числа используется функция $\text{abs}(z)$, а для вычисления фазы — $\text{angle}(z)$. Ниже даны простейшие примеры работы с комплексными числами:

```
>> i ans =
```

```
0 + 1.0000i
```

```
>> j ans =
```

```
0 + 1.0000i >> z=2+3i
```

```
z =
```

```
2.0000 + 3.0000i
```

```
>>abs(z) ans =
```

```
3.6056
```

```
>>real(z) ans =
```

```
2
```

```
>>imag(z) ans =
```

```
3
```

```
>>angle(z) ans =
```

```
0.9828
```

В MATLAB операции над числами обычно выполняются в формате, который принято считать форматом с двойной точностью. Такой формат удовлетворяет подавляющему большинству требований к численным расчетам.

2.3. Задания для лабораторной работы.

Задание 1. Найдите определитель матрицы

4	5	8	3	4
7	8	7	6	5
7	8	8	7	7
6	0	2	9	6
6	1	1	0	6

Задание 2. Перемножьте 2 матрицы, найдите $C=A*B$

A=

2	3	5	7
5	4	7	3
6	-6	8	3
6	8	1	2

B=

3	4	7	9
1	2	8	9
4	3	5	8
2	1	6	7

Задание 3. Найдите матрицу, обратную к заданной

3	4	7	9
1	2	8	9
4	3	5	8
2	1	6	7

Задание 4. Найдите график функции, заданной в полярной системе $r=a*b*\cos(c*f)$, где f - полярный угол, от 0 до $2*\pi$, в качестве a и b выберите любое число диапазоне от 50 до 200. Перевод полярной системы

в декартовую осуществляется по формуле
$$\begin{cases} x = r \cos(f) \\ y = r \sin(f) \end{cases}$$

Задание 5. Найдите значение выражения

$$y = \frac{\cos(a)^b * e^b}{(a+b)(a-b)}, \text{ где } a=5+3i, b=7-8i;$$

Задание 6.

постройте график уравнения

$$0.5x^5 - 15x^4 + 20x^3 - 20x^2 + 7x - 5 = 0$$

на интервале $-20 \dots 20$ с шагом 0.001

Определите количество корней уравнения и графически найдите каждый из них

2.4 Содержание отчета

1. Титульный лист
2. Задание на лабораторную работу
3. Листинг программы
4. Скриншот работы приложения

Лабораторная работа № 3

Пакет Matlab, синтез цифровых узлов, систем и устройств

3.1. Цель работы: Расширенные возможности Matlab для решения задач цифровой фильтрации. Практические аспекты использования среды для расчетов рекурсивных и нерекурсивных фильтров

3.2. Краткие теоретические сведения:

В реальном программировании несколько раз вы можете встретить ситуации, в которых вы хотите определить суперкласс, который объявляет некоторые методы, не предоставляя полную реализацию каждого метода. В этом случае эти суперклассы имеют имя «абстрактный». Итак, другими словами, абстрактный класс является таким классом, в котором какой-либо метод (или более одного метода) объявлен, но не определен. В этом случае метод является виртуальным с точки зрения C++ и абстрактным с точки зрения Java. Абстрактные методы не имеют реализации (body). Определение методов (creation the body) вы дадите в подклассе абстрактного класса. Если какой-то класс является абстрактным, вы не можете создать объект этого класса, другими словами этот класс не может быть создан, но должен быть унаследован.

Например, предположим, что мы хотим что-то создать: может быть автомобиль, может быть вертолет, может быть велосипед, может быть какой-то другой автомобиль. Конечно, мы можем сказать, что в общем случае вертолет, автомобиль и велосипед являются транспортным средством, и у каждого транспортного средства есть некоторые общие черты, такие как количество колес, цвет и некоторые инструкции по использованию этого автомобиля. Но, к сожалению, инструкции «идут» на велосипед, для автомобиля и для вертолета разные. Любой водитель автомобиля не может управлять вертолетом, а пилот вертолета не может управлять автомобилем. Теперь вы можете видеть, что некоторые особенности наших автомобилей распространены (например, автомобиль, велосипед и вертолет имеют цвет и количество колес), некоторые другие функции, такие как «как управлять», абсолютно разные. Как описать создание наших автомобилей с точки зрения объектно-ориентированного программирования? Эту ситуацию можно решить с помощью абстрактных классов. Предположим, что абстрактный

класс «Автомобиль» описывает общие черты автомобиля и вертолета, такие как количество колес и цвет. Также автомобиль содержит метод «go ()», чтобы описать набор инструкций для вождения автомобиля или управления вертолетом. Мы не можем предсказать, что такое подкласс класса «Транспортное средство», и мы не можем сказать, что представляет собой набор инструкций по вождению, поэтому мы можем оставить метод «go ()» пустым, без выполнения инструкций по вождению. Другими словами: сначала мы должны создать абстрактный класс «Транспортное средство», в котором содержится абстрактный метод «go ()», затем, используя наследование, мы создадим классы «Автомобиль» и «Вертолет» с использованием абстрактного класса «Транспортное средство». В классе «Автомобиль» мы реализуем метод «go ()» с использованием набора инструкций по вождению автомобиля, в классе «Вертолет» мы будем использовать другие инструкции в методе «go», которые подходят для пилотирования вертолетов. Иерархия классов объясняется на рисунке 5.1.

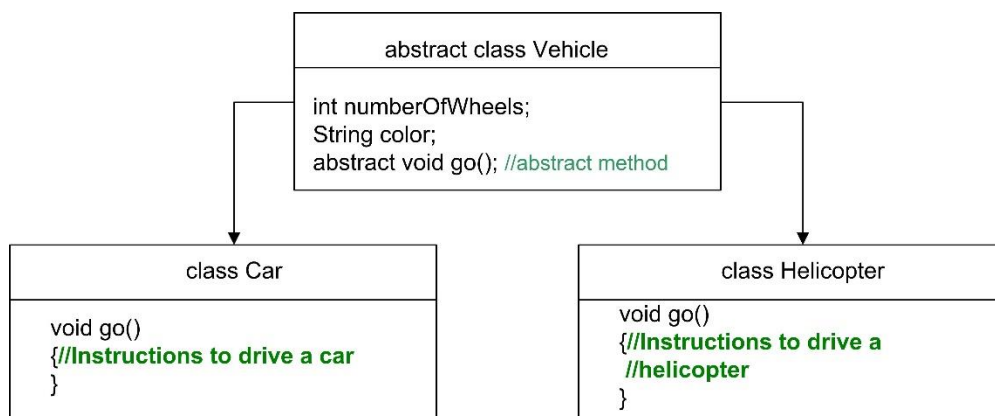


Рисунок 3.1. Иерархия классов

Программа иллюстрирует создание абстрактного класса и получение подклассов с использованием абстрактного суперкласса.

```

abstract class Vehicle//This is abstract class
{int numberOfWheels=4;
  String color;//Color of our vehicle
  abstract void go();//this is abstract method "go()"
}

class Car extends Vehicle //Class that creates a car from
vehicle
{
  void go ()
  {

```



```

        System.out.println("Set of instructions to drive a car");
    }
}

class Helicopter extends Vehicle//class that creates a
helicopter
{
    void go()
    {
        System.out.println("Here instructions to flight your
helicopter");
    }
}

public class Test{
    static public void main(String[] args)
    {Vehicle myvehicle=new Car();//Create a car
    System.out.println("Car created");
    myvehicle.go();
    System.out.println();
    myvehicle=new Helicopter();//Create a helicopter
    System.out.println("Helicopter created");
    myvehicle.go();
    }
}

```

Результатом этой программы будет:

Car created

Set of instructions to drive a car

Helicopter created

Here instructions to flight your helicopter

Теперь вы можете увидеть что-то интересное, обратите внимание на первую строку основного метода:

Транспортное средство `myvehicle = new Car ();` объект `myvehicle` является объектом класса `Vehicle`, созданного с использованием класса `Car ()`. Похоже, что `myvehicle` является «Автомобилем» «Автомобиль» в текущем случае. Затем строка `myvehicle.go ()` иллюстрирует вызов метода `go ()` объекта `myvehicle ()`; Поскольку этот объект создан как автомобиль, будет выполнен метод класса «`car`», который даст вам выход «Набор инструкций для вождения автомобиля». Затем в строке `myvehicle = new Helicopter ()` вы можете увидеть, что тот же объект «`myvehicle`» воссоздан с использованием класса «`Helicopter`». Таким образом, объект `myvehicle`

может быть автомобилем или вертолетом по вашему выбору. Эта функция называется ООР как «полиморфизм», это имя означает, что «myvehicle» имеет много форм: в нашем примере это может быть автомобиль и вертолет. Теперь вызовите метод myvehicle.go (), который вызывает выполнение метода go () класса Helicopter. Это также полиморфизм, потому что метод «go ()» иногда содержит инструкции для вождения автомобиля или иногда инструкции для вождения вертолета.

В отличие от C ++ в Java нет множественного наследования. Множественное наследование - это ситуация, когда подкласс создается с использованием более одного суперкласса. Чтобы использовать множественное наследование в Java, вы можете использовать некоторую другую функцию, называемую «Java-интерфейсы». Интерфейс выглядит как абстрактный класс, но интерфейс «абсолютно абстрактный», все методы в интерфейсе не имеют тела. Все члены данных интерфейса имеют префикс «final». Другими словами, интерфейс Java является абстрактным классом, который содержит только абстрактные методы и конечные члены данных. Следующий пример иллюстрирует создание программы, которая вычисляет квадрат прямоугольника и круга с использованием и высоты фигур. Ниже представлена иерархия классов:

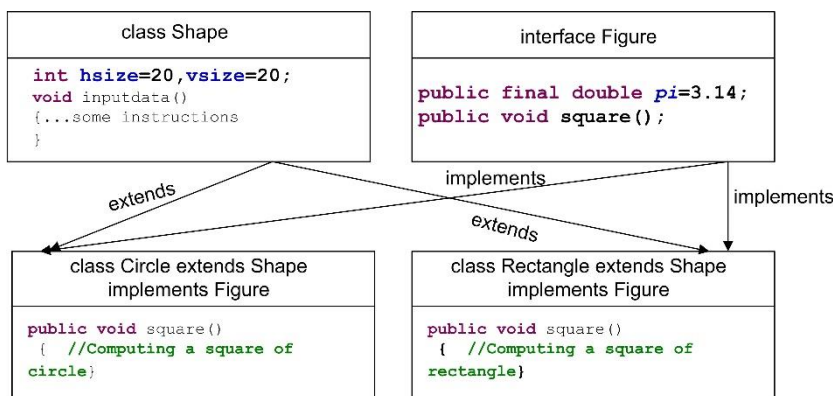


Рисунок 3.1. Иерархия классов

```

interface Figure //This is general interface for all figures
{public final double pi=3.14;
public void square();
}
  
```

```

class Shape //This is root class Shape with datamembers and
method
{
int hsize=20, vsize=20; //default size of the figure
void inputdata(int sizex, int sizey)
  {hsize=sizex; //Example of method with 2 parameters
  }
}
  
```

```

    vsize=sizey;}
}

class Circle extends Shape implements Figure
{
    public void square()//Computing a square of circle
    {double r=vsize/2;
      double result=r*r*pi;
      System.out.println("Square of the circle="+result);
    }
}

class Rectangle extends Shape implements Figure
{
    public void square()//Computing a square of rectangle
    { double result=hsize*vsize;
      System.out.println("Square of the rectangle="+result);
    }
}

public class Test{
    static public void main(String[] args)
    {Rectangle rec=new Rectangle();
      rec.square();
      Figure myfigure=new Rectangle();
      myfigure.square();
      myfigure=new Circle();
      myfigure.square();
    }
}

```

Результатом этой программы будет:

```
Square of the rectangle=400.0
```

```
Square of the rectangle=400.0
```

```
Square of the circle=314.0
```

В основной функции нашей программы мы можем создать объект «rec» класса Rectangle, затем вызывается метод square (). Чтобы использовать класс Circle, вы также можете создать объект этого класса. С другой стороны, классы Circle и Shape имеют общий интерфейс Figure, это означает, что вы можете создать один объект «myfigure» типа интерфейса «Figure» и использовать этот объект в качестве экземпляра классов Circle и Shape у них:

Figure myfigure = new Rectangle (); Эта строка означает, что объект myfigure () является прямоугольником, а метод square возвращает квадрат прямоугольника. В следующей строке кода Java

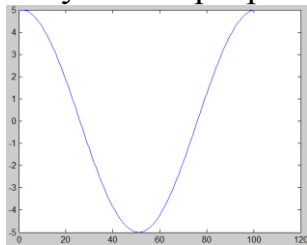
Myfigure = new Circle () объект для переназначения на новый экземпляр класса Circle. Как вы уже знаете, эта функция имеет полиморфизм имен.

3.3. Задание для лабораторной работы.

1. Создайте программный код, генерирующий синусоиду амплитудой 5 вольт и частотой 1 Гц

```
Fs=100;% это частота дискретизации  
f=1;% частота исходного сигнала  
dt=1/Fs;% dt это период дискретизации  
t=0:dt:1;% массив времени с шагом dt и длиной в 1 секунду  
x=5*cos(2*pi*t);% x- входной сигнал  
plot(x);
```

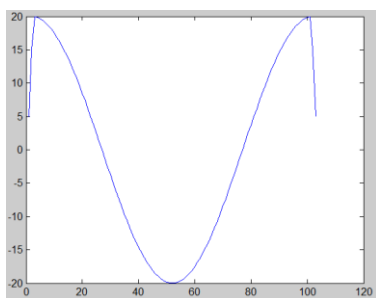
Запустите программу и посмотрите, что она выдаст:



По рисунку видно, что максимальное значение сигнала +5, минимальное -5, амплитуда определяется выражением $A=(A_{max}-A_{min})/2$ и в нашем случае составляет 5 В.

2. Добавьте к коду п.1 программу, которая будет проводить фильтрацию сигнала используя цифровой фильтр $y[n]=x[n]+2x[n-1]+x[n-2]$; и посмотрите на результат

```
h=[1 2 1;% импульсный ответ фильтра  
y=conv(x,h); % свертка фильтра и исходного сигнала  
figure;plot(y); % выходной сигнал
```



Видно, что максимальное значение сигнала на выходе 20, минимальное -20, амплитуда составляет 20 В. Таким образом, передаточный коэффициент для частоты 1 Гц равняется:
 $k=A_{вых}/A_{вх}=20/5=4$

3. Измените систему. Используйте уравнение: $y[n]=2x[n]-x[n-1]-2x[n-2]$; Постройте график входного и выходного сигнала и определите коэффициент передачи для частоты 10 Гц.

4. Изменяя частоту сигнала (строка $f=10$ в программе), заполните таблицу

Частота, Гц	10	20	30	40	50	60	70	80	90
Выходной сигнал, В									
Коэффициент передачи, К									

5. Постройте график этой таблицы. По оси абсцисс отложите частоту, по оси ординат - коэффициент передачи. Назовите полученный рисунок "Функция передачи системы"

6. Измените в программе строку, в которой задается исходный сигнал:
 $x=5*\cos(2*\pi*t*f);$

на $x=5*\cos(2*\pi*t*1)+2*\sin(2*\pi*t*5);$

Запустите программу и зарисуйте (скопируйте с экрана) форму входного сигнала и выходного сигнала.

3.4 Содержание отчета

1. Титульный лист
2. Задание на лабораторную работу
3. Листинг программы
4. Скриншот работы приложения