Министерство образования Республики Беларусь Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

# Отчёт

по лабораторной работе №10

по дисциплине ***«Объектно-ориентированное программирование»***

тема: «Java-пакеты и секции импорта. Архитектурный шаблон проектирования MVC. Линейные алгоритмы»

Название бригады: «**Best progger**»

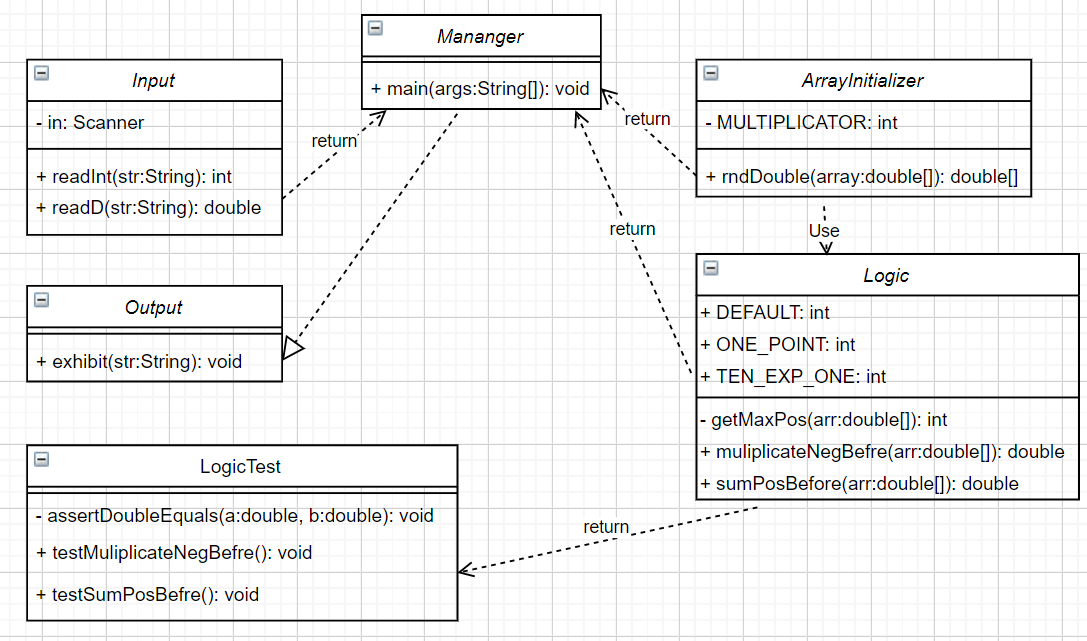
Исполнитель: студенты группы 10701118

Воробей Иван Александрович

Преподаватель: Иванченко Виктор Викторович

2018-2019 учебный год

## ЛАБОРАТОРНАЯ РАБОТА №10



*ПРИЛОЖЕНИЕ А*

}

#### 

**Контрольные вопросы**

1) Что такое массив?

An array is a type of store data, witch contain similar type items. Each item has own serial number.

2) Зачем нужно при проектировании и реализации программы использовать

массивы?

We use arrays in order to get fast access to content. As it effective to store data.

3) Какие существуют типы массивов в Java?

There are one-dimensional and multidimensional.

4) Чем массивы в Java отличаются от массивов в C/C++/C#?

In java it’s reference. In C it’s primitive.

5) Опишите особенности Java-массивов и его свойства.

It’s a reference object.

6) Почему в Java массивы занимают особое место по сравнению с другими контейнерами? Опишите основные преимущества и недостатки Java-массивов

по сравнению с существующими библиотечными контейнерами: эффективность, типизация и возможность хранить примитивы.

Fast access. Can store primitives.

7) Каково время доступа к элементам массива?

O=1;

8) Почему нумерация элементов массива начинается с нуля?

Because it faster. Address =addressOfFirstItem+sizeOf(type)\*serialNumber

9) Какие существуют нотации в Java объявления переменной типа массива?

dataType[] arrayName;

dataType[] aName;

10) Способы объявления, конструирования и инициализации массивов в Java?

**int**[] myArray; // declaration

myArray = **new** int[10];//memory allocation

**int**[] myArray = **new** int[10];

When we use new all array is filled up by nils.

int[] myArray ;

myArray=new int[] {4,3};

int[] my= {3,2};

int[] yours=new int[] {2,34,4};

11) Если массив – это объект, а все объекты создаются на базе описанного класса, то где хранятся классы-массивов?

Массив в Java – это объект, класс которого создаётся «на лету» перед созданием объекта массива во время выполнения кода.

12) Как передать массив в качестве входного параметра в метод?

Public static void sort(int …ms)

Public static void sort(int[] args)

Public void sort(int args[])

13) Как возвратить из метода значение в виде массива?

Public int[] method(){return new[5];}

Public int method()[]{return new[5];}

14) Как инициализировать элементы одномерного массива во время объявления ссылочной переменной?

Look at tenth question.

15) Как инициализировать элементы одномерного массива, который создаётся

«на лету» в виде анонимного объекта и либо передаётся в качестве параметра в вызываемый метод, либо возвращается как результат работы метода?

i=Logic.*getMaxPos*(new double [4]);

16) Массивы и примитивные типы данных.

17) Массивы и ссылочные типы данных.

18) Что физически содержит (хранит) внутри себя массив?

Array consist of addresses for each element in memory. They stay subsequently.

19) Какие могут возникнуть исключения при работе с массивами и на что они

указывают?

ArrayIndexOutOfBoundsException

20) Какие существуют способы копирования (клонирования) массивов (элементарных и ссылочных типов)?

Object [] ob=new Object[4];

Object [] ob1=new Object[4];

ob1=ob;

int[] my =new int[] { 2, 3, 4 };

int[] yours = new int[] { 2, 3, 4 };

my=yours;

In example above the reference is copied. Adresses are the same.

In example below the new object is created. Adresses are different.

int[] my =new int[] { 2, 3, 4 };

int[] yours = new int[] { 1, 1, 1 };

yours=my.clone();

System.***out***.println(my.equals(yours));//compare references

System.***out***.println(my == yours);//compare references

21) Что такое поверхностное и глубокое копирование объектов?

Точной копией оригинала является его клон. В Java это означает возможность создавать объект с аналогичной структурой, как и у исходного объекта. Метод clone() обеспечивает эту функциональность. Поверхностное копирование копирует настолько малую часть информации, насколько это возможно. По умолчанию, клонирование в Java является поверхностным, т.е. Object class не знает о структуре класса, которого он копирует. При клонировании, JVM делает такие вещи:

1. Если класс имеет только члены примитивных типов, то будет создана совершенно новая копия объекта и возвращена ссылка на этот объект.
2. Если класс содержит не только члены примитивных типов, а и любого другого типа класса, тогда копируются ссылки на объекты этих классов. Следовательно, оба объекта будут иметь одинаковые ссылки.

Глубокое копирование дублирует все. Глубокое копирование — это две коллекции, в одну из которых дублируются все элементы оригинальной коллекции. Мы хотим сделать копию, при которой внесение изменений в любой элемент копии не затронет оригинальную коллекцию.

22) Что такое поверхностное и глубокое копирование массивов?

23) Для чего используется утилитный класс java.util.Arrays?

It is used for help programmer to process the arrays.

24) Как сравнить два массива?

System.***out***.println(my.equals(yours));//compare references

System.***out***.println(my == yours);//compare references

System.***out***.println(Arrays.*equals*(yours,my));//Compare content

25) Как отсортировать элементы массивов?

Arrays.*sort*(yours);

26) Как произвести поиск элемента в массиве?

i=Arrays.*binarySearch*(yours, 9);

System.***out***.println(i);

27) Как объявить метод с переменным количество передаваемых параметров?

Как реализована логика данного метода?

public static int getMaxPos(double... arr) {}

28) Каковы основные правила объявления методов с переменным количеством

параметров?

public static int getMaxPos(int u,double ...arr)

It should be the last argument.

29) Как использовать расширенный цикл for (ещё его иногда называют for-each),

который появился с JDK 5.0?

For each loop. It’s kind of for loop. It contain a variable, that get value of each element of massive or other type of data. With each new iteration new value is assigned to that variable.

30) Чего нельзя сделать с помощью расширенного цикла for в Java?

You can’t go through array from end to start.

We can’t change iterative variable.