**Санкт-Петербургский национальный исследовательский университет**

**информационных технологий, механики и оптики**

**Кафедра информатики и прикладной математики**

Программирование Интернет-Приложений

Лабораторная работа №1

Выполнил: Гхази Даниэль

Группа P3218

Преподаватель: Гаврилов А.В.

2016 г.

**Текст Задания**

На языке Java написать простейшую консольную программу, выводящую в поток System.out список переданных ей аргументов командной строки. Дополнить эту программу использованием любого из классов стандартной библиотеки или любой синтаксической конструкции языка Java.

**Текст Программы**

**public class** Main {  
  
 **public static void** main(String[] args)  
 {  
 **if**(args.**length** > 0)  
 **for** (String arg: args)  
 System.***out***.println(arg);  
 **else** System.***out***.println(**"No arguments input."**);  
  
 **int**[] array = {3, 2, 1};  
 java.util.Arrays.*sort*(array);  
  
 **for**(**int** elem: array)  
 System.***out***.println(elem);  
 }  
}

**Описание:**

Программа принимает аргументы командной строки и выводит их в поток вывода. Далее происходит сортировка массива целых чисел с помощью метода sort стандартного класса Arrays.

**Описание утилит в составе JDK:**

**javac** - Компилятор исходные тексты (файлы с расширением .java) в байт-код (файлы с расширением .class).

В одном файле с расширением .java должен находиться только один public-класс и имя этого класса (без имени пакета) должно совпадать с именем этого файла (без расширения).

**Формат:**

javac [параметры] файлы

**Основные параметры:**

*-classpath* путь

Переопределяет путь поиска классов, заданной переменной CLASSPATH

*-d каталог*

Задает каталог для хранения классов (по умолчанию используется текущий каталог). Файлы классов размещаются в подкаталогах в соответствии с именами пакетов классов.

*-deprecation*

Устанавливает режим выдачи сообщения при каждом использовании устаревшего API.

-verbose

Устанавливается режим выдачи сообщений о ходе компиляции.

**java** - Интерпретатор байт-кода. Запускает Java-программы (файлы с расширением .class). Программа, которую необходимо выполнить должна представлять собой класс с именем имя\_класса (без расширения .class, но с указанием пакета, которому принадлежит класс) и содержать метод main() с описанием:

public static void main(String args[])

Аргументы, указанные в командной строке, помещаются в массив args[] и передаются методу main.

**Формат:**

java [параметры] имя\_класса [аргументы]

**Основные параметры:**

*-classpath* *путь*

Переопределяет путь поиска классов, заданный переменной CLASSPATH.

*-jar*

Выполняет приложение, упакованное в архив.

*-verbose*

Устанавливается режим выдачи сообщений о загрузке классов.

*-dимя=значение*

Присваивает системному свойству с заданным именем указанное значение

**javadoc** - Создает документацию в форате HTML для указанных пакетов или файлов исходных текстов Java. Данные для документирования берутся из комментариев для документации, имеющих вид /\*\* комментарий \*/, в которых могут исплользоваться форматирующие метки HTML.

**Формат:**

javadoc [параметры] файлы

javadoc [параметры] пакет

**Основные параметры:**

*-classpath путь*

Переопределяет путь поиска классов, заданный переменной CLASSPATH.

*-d каталог*

Задает каталог для записи документации.

*-docecoding кодировка*

Задает кодировку символов для документации.

*-encoding кодировка*

Задает кодировку символов для исходных текстов.

*-author*

Включает в документацию информацию об авторе.

*-version*

Включает в документацию информацию о версии.

*-verbose*

Устанавливается режим выдачи сообщений о ходе документирования

**jar** - Используется для создания архивных файлов Java (JAR) и работы с ними. JAR-файл представляет собой сжатый ZIP-файл с дополнительным файлом описания.

Параметры командной строки jar задаются в виде блока записанных слитно букв, которые передаются в виде одного аргумента, а не через отдельные аргументы командной строки. Первая буква такого аргумента задает необходимое действие, которое должна выполнить программа jar. Остальные буквы в этом аргументе являются необязательными. Различные аргументы файлов зависят от того, какие буквы параметров заданы.

**Формат:**

jar c|t|x|i|u[f][m][v] [jar-файл] [файл описания] [файлы]

Первым аргументом командной строки jar является набор символов, задающих операцию, которая должна быть выполнена. Первый символ определяет основную операцию и является обязательным.

Возможны следующие варианты:

*c* – создать новый jar-файл

*t* – вывести список файлов, содержащихся в архиве

*x* – извлечь содержимое jar-файла

*u* – обновить jar-файл

*i* – сгенерировать файл INDEX.LIST, содержащий информацию о jar-файле, такую как зависимые jar-файлы и расположение всех пакетов

Вслед за идентификатором, определяющим выполняемое действие, могут следовать необязательные параметры:

*f* - указывает на то, что имя JAR-файла, который необходимо создать, из которого нужно извлечь файлы или получить список содержащихся файлов, задается в командной строке.

*m* - используется только в сочетании с параметром с и указывает на то, что jar должна читать файл описания, указанный в командной строке и использовать его в качестве основы для создания описания, которое включается в JAR-файл.

*v* - если этот параметр задается вместе с параметром с, то выводится имя каждого добавляемого в архив файла со статистикой его сжатия.

**Вывод:**

В результате выполнения лабораторной работы я ознакомился с синтаксисом языка Java, использованием методов стандартных классов и приемом аргументов командной строки. Так же мною были изучены утилиты jdk, такие как java и javac. В дальнейшем эти знания можно использовать для написания более сложных программ.