**Санкт-Петербургский национальный исследовательский университет**

**информационных технологий, механики и оптики**

**Кафедра информатики и прикладной математики**

Программирование интернет-приложений

Лабораторная работа №1

«Определение длины световой волны при помощи опыта Юнга».

Выполнил Кудряшов А.А.

Группа 2121

2012 г.

## Часть 1

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.io.InputStreamReader;

**public** **class** HelloWorld {

**static** **boolean** Question\_H\_W() {

**try** {

BufferedReader b = **new** BufferedReader(**new** InputStreamReader(System.*in*));

System.*out*.print("4 + 2 = ");

String a = b.readLine();

**if** (a.equals("6"))

**return** **true**;

**else** {

**return** **false**;

}

} **catch** (IOException e) {

e.printStackTrace();

**return** **false**;

}

}

**public** **static** **void** main(String args[]) **throws** IOException {

**if** (*Question\_H\_W*()) {

System.*out*.print("Hello World!");

} **else** {

System.*out*.print("Попробуйте вновь");

}

}

}

# Часть 2

## Javac

Компилирует исходные тексты (файлы с расширением .java) в байт-код (файлы с расширением .class).

javac [ параметры ] файлы

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

Параметры:

-classpath *путь*

Переопределяет путь поиска классов, заданный 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.

-nowrite

Устанавливает режим проверки, при котором откомпилированный класс

не записывается.

-O

Разрешает использовать оптимизацию классов.

-verbose

Устанавливается режим выдачи сообщений о ходе компиляции.

# Java

Интерпретатор байт-кода. Запускает Java-программы (файлы с расширением.class).

java [ *параметры* ] *имя\_класса* [ *аргументы* ]

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()

Параметры:

-classpath *путь*

Переопределяет путь поиска классов, заданный переменной CLASSPATH.

-cs (-checksource)

Указывают на необходимость сравнения времени модификации файла

класса и исходного текста и при необходимости перекомпиляции

программы.

-D*имя=значение*

Присваивает системному свойству с заданным именем указанное значение.

-noverify

Отключает режим проверки байт-кода.

-verify

Включает режим проверки байт-кода для всех классов (по умолчанию

проверяются только классы, загруженные по сети).

-verbose

Устанавливается режим выдачи сообщений о загрузке классов.

# javadoc

Создает документацию в формате HTML для указанных пакетов или файлов

исходных текстов Java.

javadoc [ *параметры* ] *файлы*

javadoc [ параметры ] пакет

Данные для документирования берутся из комментариев для документации,

имеющих вид /\*\* комментарий \*/, в которых могут использоваться

форматирующие метки HTML.

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.

Документирующие комментарии

/\*\*

\* Первое предложение является кратким описанием класса или

\* метода. Далее следуют более подробное дополнительное

\* описание. После описаний могут располагаться специальные

\* теги, обеспечивающие дополнительное форматирование.

\* Для классов могут употребляться следующие теги:

\* @author автор

\* @version версия

\* @see класс

\* @see пакет.класс#метод

\* @deprecated объяснение

\*/

public class myClass {

/\*\*

\* Эта переменная содержит очень полезную информацию.

\* Комментарии должны предшествовать объявлению класса,

\* метода или переменной.

\* Для переменных используются:

\* @see класс

\* @see класс#метод

\* @deprecated объяснение

\*/

public int myVariable;

/\*\*

\* Метод, устанавливающий все необходимые значения.

\* Для методов, кроме тегов, используемых для переменных,

\* могут также использоваться:

\* @param параметр описание

\* @return описание

\* @exception исключение описание

\*/

public String myMethod(int a, int b) throws myException {

return str;

}

}

# Appletviewer

Используется для просмотра апплетов, ссылки на которые имеются в HTML-

документах.

appletviewer [ *параметры* ] *url/файлы*

Указанные HTML-файлы загружаются, и все апплеты, на которые в них

имеются ссылки в виде тега <APPLET>, отображаются каждый в собственном

окне.

Параметры:

-debug

Апплет запускается под управлением отладчика.

-J*аргумент*

Передает указанный аргумент командной строки интерпретатору.

-encoding *кодировка*

Задает кодировку символов для HTML-документов.