

Цель: научиться использовать утилиты командной строки для компиляции и запуска приложений и апплетов

Предварительная подготовка:

1. Должен быть установлен JDK
2. Должны быть прописаны пути до подкаталога bin в переменную окружения Path . Например: D:\Java\jdk1.7.0_40\bin
3. Каталоги docs и src из архива docs-src.zip должны быть разархивированы, например, в каталог D:\Java\Archive
4. На диске должен быть создан каталог с именем Experiments, например, D:\Experiments

Задача 1 - компиляция и запуск простого приложения на языке Java. Выполните следующие действия.

1. Компиляция и запуск простого приложения на языке Java

1.1. Скопируйте файл HelloWorld.java из каталога [Archive\src\Lab_1](#) в каталог Experiments

1.2. Запустите командную строку (cmd в Windows, или terminal в Linux) и установите текущим каталогом

D:\Experiments :

d: <ввод>

cd D:\Experiments

- 1.3. Выполните команду из командной строки в текущем каталоге:

javac HelloWorld.java

В результате успешного выполнения этой команды в том же каталоге должен появиться файл с именем HelloWorld.class. Расширение class указывает, что это исполняемый (бинарный) файл Java.

1.4. Запустите на исполнение следующей командой (обратите внимание, что расширение не пишется):

java HelloWorld

В результате успешного выполнения этой команды на экране должно напечататься сообщение:

Hello World!

Задача 2 - Компиляция и запуск апплета

2. Компиляция и запуск апплета

2.1. Скопируйте файлы HelloWorldApplet.java и AppletHTM.html из каталога Archive\src в каталог Experiments

2.2. Скомпилируйте файл HelloWorldApplet.java аналогично файлу HelloWorld.java

2.3. Протестируйте работу апплета с помощью утилиты appletviewer (описано в лекции, п.1.9.2). Сравните результат работы с рисунком 1.5

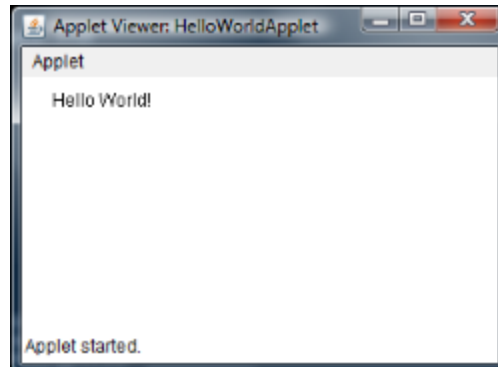


Рисунок 1.5 - Результат работы. Окно утилиты appletviewer

2.4. Протестируйте работу апплета с помощью браузера. Откройте AppletHTM.html из каталога Experiments. Сравните работу апплета с рисунком 1.6

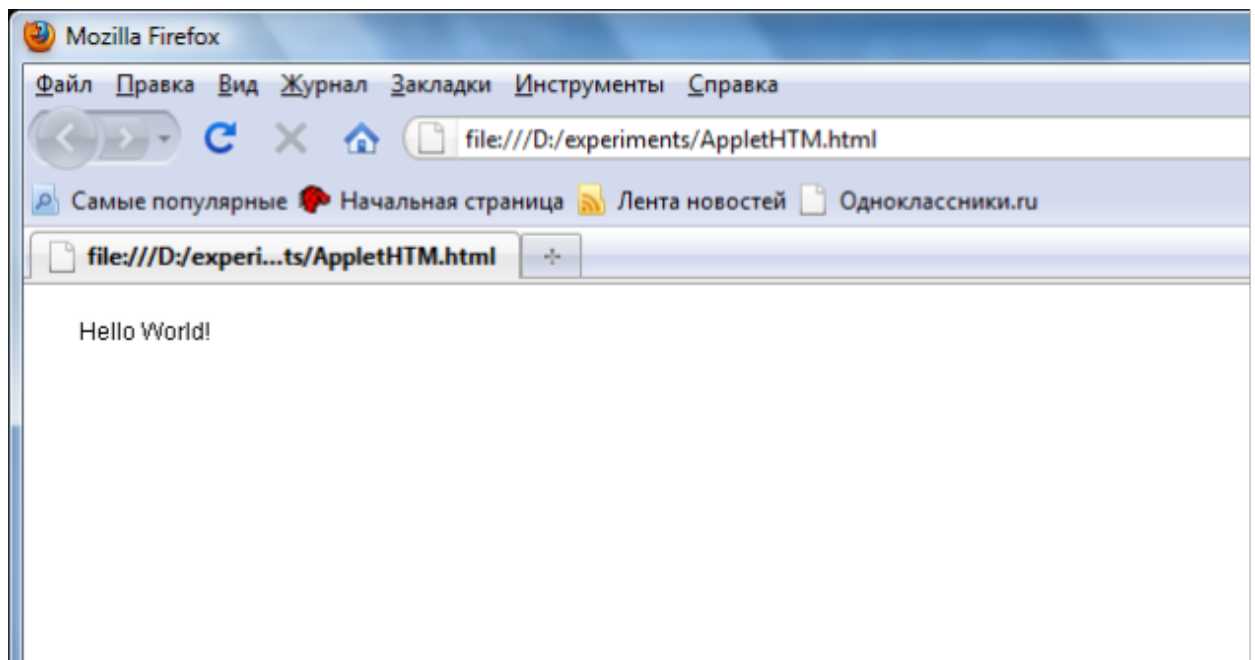


Рисунок 1.6 - Результат работы. Окно браузера, отобразившего апплет

Задача 3 - Исследование реакции компилятора на ошибки в программе.

3. Исследование реакции на ошибки в исходном коде

3.1. Внесем ошибку в код первой программы на Java (листинг 1.1). В первой строке вместо имени класса HelloWorld введем имя Hello. Сохраним файл, и попробуем выполнить компиляцию из командной строки (см. п 1.3 Задачи 1).

```
HelloWorld.java:2: class Hello is public, should be declared in a file named Hello.java
public class Hello {
^
1 error
```

Рассмотрим это сообщение. В первой строке сообщается номер строки исходного кода, в которой обнаружена ошибка (HelloWorld.java:2). В данном случае - это строка 2. Далее выводится описание ошибки. Мы поменяли имя класса, а по правилам Java все классы с модификатором public должны объявляться в файлах с таким же именем. Поэтому компилятор нам сообщает, что класс Hello должен тогда быть объявлен в отдельном файле с именем Hello.java. В следующей строке сообщения цитируется строка исходного кода с ошибкой. А в следующей строке символом ^ указывается позиция ошибочного символа. Не следует интерпретировать это слишком буквально: что именно слово class является ошибочным. Нужно прочитать описание ошибки в первой строке, именно описание дает наиболее ясное понимание причины, и затем делать выводы. Чтобы исправить ошибку есть две возможности: 1- вернуть начальное название класса, и 2 - переименовать файл. Воспользуйтесь второй возможностью. Переименуйте файл как требует компилятор. Убедитесь, что после переименования, компиляция и запуск проходят успешно.

3.2. Внесем ошибку, связанную с отсутствием точки запятой после оператора. Удалите точку с запятой в строке вызова System.out.println(); Сохраните файл, скомпилируйте. Должно быть выдано сообщение

```
D:\experiments>javac Hello.java
Hello.java:4: ';' expected
System.out.println("Hello World!!!")
^
1 error
```

Здесь аналогично предыдущему пункту дано имя исходного файла, номер строки, описание ошибки, цитата, маркер позиции ошибки и общее количество обнаруженных ошибок. Пока не исправляйте ошибку. Перейдите к следующему пункту лабораторной.

3.3. Внесем в этот же код еще одну ошибку. Удалите одну закрывающую фигурную скобку в самом конце исходного кода. Выполните компиляцию. Просмотрите внимательно сообщение об ошибках.

Исправьте все ошибки, выполните компиляцию, убедитесь в правильной работе программы.