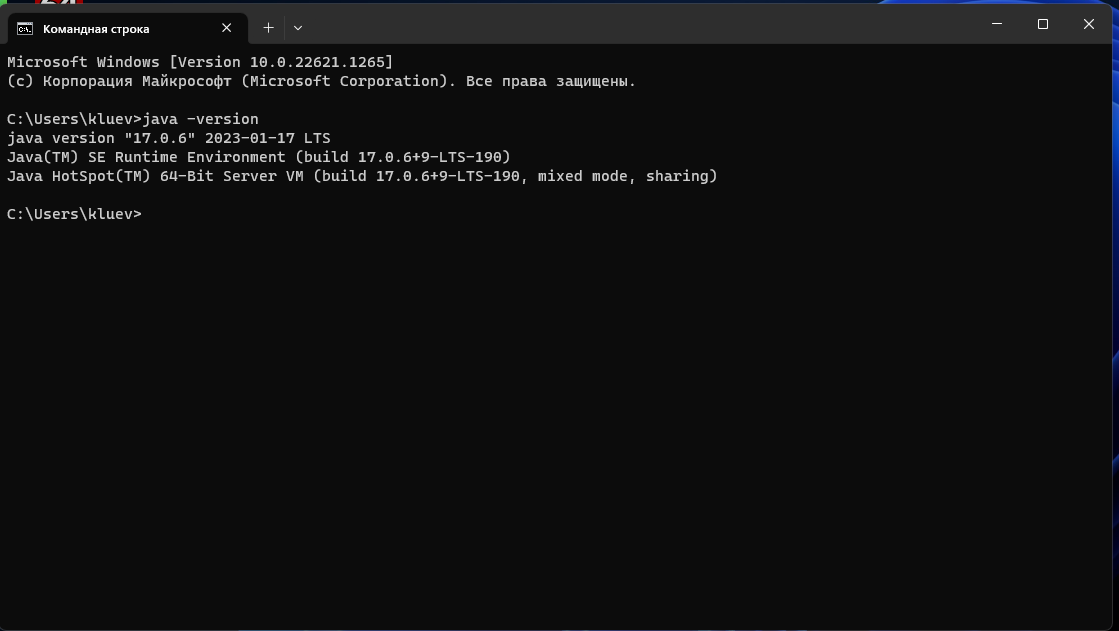
Лабораторная работа #1

«Знакомство с языком программирования Java»

Цель: получить представление о написание программ на языке программирования Java с использованием командной строки, интегрированной среды разработки, а также решение задач на сайте timus.

Ссылка на GitHub: https://github.com/Cherberushka/sber\_java

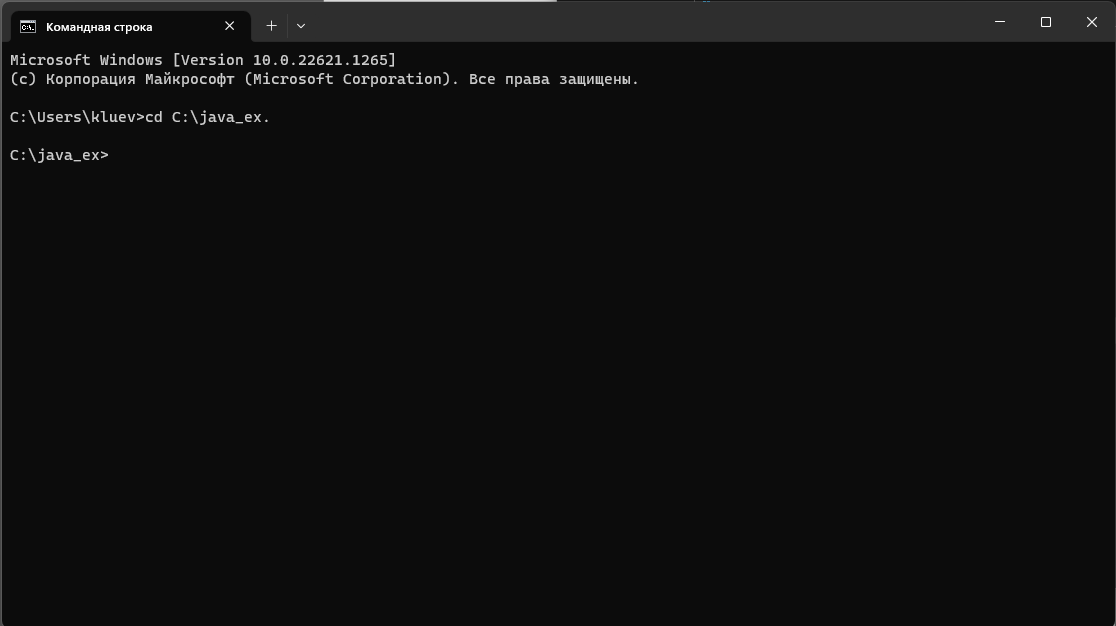
# Установка необходимого программного обеспечения

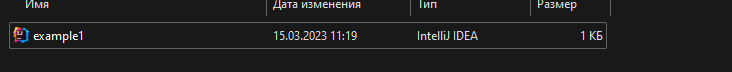


# Компиляция и запуск java программ из командной строки

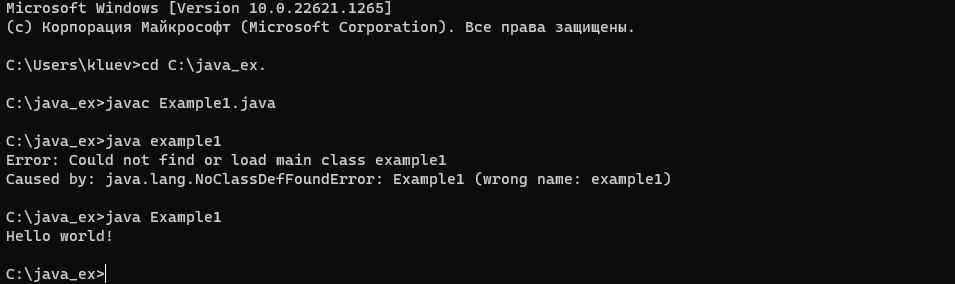
## Вывод на консоль

Создаем папку «java\_ex» в корне диска C, и затем создаем файл java.



****

Вводим в cmd javac Example1.java, компилятор создает в каталоге скомпилированный файл «example1.class». Запустим программу в командной java Example1.

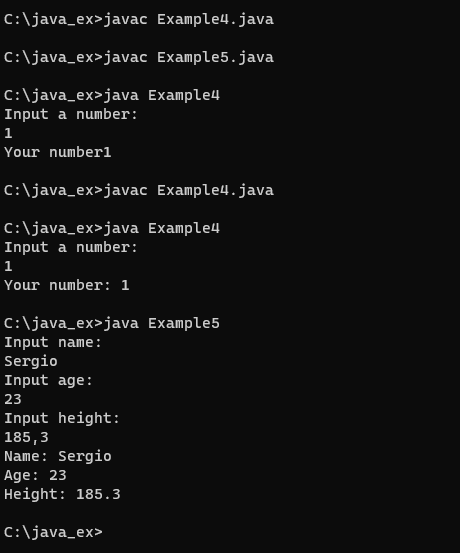
****

Создадим файлы «example2» и «example3», после чего скомпилируем их и запустим созданные файлы.

|  |  |
| --- | --- |
|  |  |

****

## Ввод с консоли



# Вопросы для закрепления материала

1. Будет ли исполняться программа без ошибок. Если без ошибок, то почему все работает. Если есть ошибка, то указать:

class SomeScopeExample {  
 public void eatIfHungry(boolean hungry) {  
 if (hungry) {  
 int bitesOfCheese = 1;  
 {  
 var teenyBit = true;  
 System.*out*.println(bitesOfCheese);  
 }  
 }  
 System.*out*.println(teenyBit);  
 }  
}

Ответ. При запуске программы компилятор выдаст ошибку java: cannot find symbol.

Ошибка в том, что переменная "teenyBit" объявлена в блоке кода, ограниченном фигурными скобками, и не может быть использована за их пределами. Чтобы исправить ошибку, нужно объявить переменную "teenyBit" внутри метода "eatIfHungry" вне блока кода, где она нужна.

class SomeScopeExample {  
 public void eatIfHungry(boolean hungry) {  
 boolean teenyBit = false; // объявляем переменную здесь  
 if (hungry) {  
 int bitesOfCheese = 1;  
 {  
 teenyBit = true; // изменяем значение переменной здесь  
 System.*out*.println(bitesOfCheese);  
 }  
 }  
 System.*out*.println(teenyBit);  
 }  
}

1. Даны два куска кода. Определить какой из кусков будет работать без ошибок, а какой даст ошибку. В куске, который дает ошибку, объяснить почему:

|  |  |
| --- | --- |
| short bird = 'd'; char mammal = (short)83; char mammal = 83; | short reptile = 65535; char fish = (short)-1; char fish = -1; |

Ответ. Первый кусок кода будет работать без ошибок. В нем создаются переменные типа short и char, значения которым присваиваются корректно.

Второй кусок кода даст ошибку

1) Из-за того, что число 65535 не может быть присвоено переменной типа short, потому что она принимает значения от -32768 до 32767

2) Из-за приведения значения -1 к типу char. Тип данных char хранится в 2 байтах (16 бит), а диапазон состоит только из положительных чисел от 0 до 65 535.

1. Почему код ниже не будет компилироваться и как это исправить

long max = 3123456789;

-->integer number too large

Java жалуется, что число вне допустимого диапазона, так как Java рассматривает его как int.

Один из способов исправить это — просто добавить символ L к числу.

long max = 3123456789L;

1. Что из нижеперечисленного будет компилироваться, а что нет. И почему:
2. int value = null;
3. List<Integer> heights = new ArrayList<>();  
   heights.add(null);
4. String s = null;

1) Не скомпилируется, так как тип данных int не может принимать значение null.

2) Будет компилироваться, так как тип данных List принимает объекты и значение null является допустимым объектом.

3) Будет компилироваться, так как тип данных String может принимать значение null.

1. Что из нижеперечисленного верно? (Выберите все, что применимо)

A. Классовая переменная типа "boolean" имеет значение по умолчанию 0

B. Переменная класса типа boolean по умолчанию имеет значение false.

C Переменная класса типа boolean по дефолту равна null.

D. Переменная класса типа long по умолчанию имеет значение null

E. Переменная класса типа long по умолчанию имеет значение 0L

F. Классовая переменная типа long имеет значение по умолчанию 0

G. Ничего из вышеперечисленного

1. Что из нижеперечисленного верно? (Выберите все, что применимо).

A. Локальная переменная типа boolean по умолчанию равна null.

B. Локальная переменная типа float по умолчанию имеет значение 0.0f.

C. Локальная переменная типа double по умолчанию равна 0.

D. Локальная переменная типа Object имеет значение по умолчанию null.

E. Локальная переменная типа boolean по умолчанию имеет значение false.

F. Локальная переменная типа float по умолчанию равна 0.0.

G. Ничего из вышеперечисленного

1. Что из нижеперечисленного верно? (Выберите все, что применимо)

A. Переменная экземпляра типа float по умолчанию равна 0.

B. Переменная экземпляра типа char по умолчанию равна null.

C. Переменная экземпляра типа double имеет значение по умолчанию 0.0.

D. Переменная экземпляра типа int по умолчанию равна null.

E. Переменная экземпляра типа String имеет значение по умолчанию null.

F. Переменная экземпляра типа String по умолчанию принимает значение пустой строки "".

G. Ничего из вышеперечисленного

1. Какие из следующих утверждений о фрагменте кода являются верны? (Выберите все, что применимо.)

1. short numPets = 5L;  
2. int numGrains = 2.0;  
3. String name = "Scruffy";  
4. int d = numPets.length();  
5. int e = numGrains.length;  
6. int f = name.length();

1. Строка 1 генерирует ошибку компиляции.

2. Строка 2 генерирует ошибку компиляции.

3. Строка 3 генерирует ошибку компиляции.

4. Строка 4 генерирует ошибку компиляции.

5. Строка 5 генерирует ошибку компиляции.

6. Строка 6 генерирует ошибку компиляции.

1. Какой из следующий наименований переменных корректный?

A. \_

B. \_helloWorld$

C. true

D. java.lang

E. Public

F. 1980\_s

G. \_Q2\_

1. Написать в строках с System.out.println() что выведется в консоль: true или false

String firstString = "new";  
String secondString = "new";  
System.*out*.println(firstString == secondString); //false  
System.*out*.println(firstString.equals(secondString)); //true  
String thirdString = new String("new");   
System.*out*.println(thirdString.equals(secondString)); //true  
System.*out*.println(thirdString == secondString); //false  
System.*out*.println(thirdString.intern() == firstString); //true  
System.*out*.println(thirdString.equals(secondString)); //true

1. В следующем задании: 1. Определить есть ли ошибки в коде. Если есть, то где и почему. Исправить при наличии 2. Написать цикл, который будет итерироваться по массиву (по поводу цикла: for и forEach)

1) int[] intArray = {3, 4, 5};  
2) float[] floatArray = float[] {7.2f, 1.8f, 9.3};  
3) char[] charArray = new char[2];

1. В строке 1, лишний "float[]" - массив уже указан в объявлении переменной, так что правильный код должен выглядеть так: "float[] floatArray = {7.2f, 1.8f, 9.3f};"

2. В строке 3, массив символов создается с размерностью 2, но не задает какие именно символы должны быть в этом массиве. Поэтому после создания массива нужно добавить символы.

Пример корректный кода.

int[] intArray = {3, 4, 5};  
float[] floatArray = {7.2f, 1.8f, 9.3f};  
char[] charArray = new char[2];  
charArray[0] = 'a';  
charArray[1] = 'b';  
System.*out*.println(charArray);

Перебор массива через for

int[] intArray = {3, 4, 5};  
  
for (int i = 0; i < intArray.length; i++) {  
 String strToPrint = "intArray[" + i + "]=" + intArray[i];  
 System.*out*.println(strToPrint);  
}