ЛАБОРАТОРНАЯ РАБОТА № 6

«Введение в классы, часть 2»

Выполнил Козлов И.А.   
Группа РИВ-120908у

**Цель работы**  
Введение в работу с классами Java.

**Git репозиторий**

https://github.com/Catlerc/javadev

**Задание 1**

Напишите программу с классом, в котором есть два поля: символьное и текстовое. В классе должен быть перегруженный метод для присваивания значений полям. Если метод вызывается с символьным аргументом, то соответствующее значение присваивается символьному полю. Если метод вызывается с текстовым аргументом, то он определяет значение текстового ноля. Методу аргументом также может передаваться символьный массив. Если массив состоит из одного элемента, то он определяет значение символьного поля. В противном случае (если в массиве больше одного элемента) из символов массива формируется текстовая строка и присваивается значением текстовому полю.

**Результат**

class Stats {  
 static public int[] stats(int[] ints) {  
 var max = Integer.*MIN\_VALUE*;  
 var min = Integer.*MAX\_VALUE*;  
 for (int i = 0; i < ints.length; i++) {  
 max = Math.*max*(max, ints[i]);  
 min = Math.*min*(min, ints[i]);  
  
 }  
 return new int[]{max, min};  
 }  
}  
  
public class Task10 {  
 public static void main(String[] args) {  
 System.*out*.println(Arrays.*toString*(Stats.*stats*(new int[]{1, 2, 3, 4, 5})));  
 }  
}

**Задание 2**

Напишите программу с классом, в котором есть закрытое статическое целочисленное ноле с начальным нулевым значением. В классе должен быть описан статический метод, при вызове которого отображается текущее значение статического поля, после чего значение поля увеличивается на единицу.

**Результат**

class Counter {  
 static private int *value*;  
  
 static public void printAndInc(){  
 System.*out*.println(*value*);  
 *value*++;  
 }  
}  
  
public class Task2 {  
 public static void main(String[] args) {  
 Counter.*printAndInc*();  
 Counter.*printAndInc*();  
 Counter.*printAndInc*();  
 Counter.*printAndInc*();  
 }  
}

**Задание 3**

Напишите программу с классом, в котором есть статические методы, которым можно передавать произвольное количество целочисленных аргументов (или целочисленный массив). Методы, на основании переданных аргументов или массива, позволяют вычислить: наибольшее значение, наименьшее значение, а также среднее значение из набора чисел.

**Результат**

class SimpleMath {  
 static int max(int... args) {  
 var tmp = Integer.*MIN\_VALUE*;  
 for (int arg : args) {  
 tmp = Math.*max*(tmp, arg);  
 }  
 return tmp;  
 }  
  
 static int min(int... args) {  
 var tmp = Integer.*MAX\_VALUE*;  
 for (int arg : args) {  
 tmp = Math.*min*(tmp, arg);  
 }  
 return tmp;  
 }  
  
 static int avg(int... args) {  
 var tmp = 0;  
 for (int arg : args) {  
 tmp += arg;  
 }  
 return tmp / args.length;  
 }  
}  
  
public class Task3 {  
 public static void main(String[] args) {  
 System.*out*.println("max(1,2,3):" + SimpleMath.*max*(1, 2, 3));  
 System.*out*.println("min(1,2,3):" + SimpleMath.*min*(1, 2, 3));  
 System.*out*.println("avg(1,2,3):" + SimpleMath.*avg*(1, 2, 3));  
 }  
}

**Задание 4**

Напишите программу, в которой описан статический метод для вычисления двойного факториала числа, переданного аргументом методу. По определению, двойной факториал числа п (обозначается как n!!) — это произведение через одно всех чисел, не больших числа п. То есть n!! = n \* (n - 2) \* (n - 4)\* ... (последний множитель равен 1 для нечетного п и равен 2 для четного n). Например, 6!! = 6 х 4 х 2 = 48 и 5!! = 5 х 3 х 1 = 15.

**Результат**

class Factorial {  
 static public long doubleFactorial(int n) {  
 long tmp = 1;  
 for (int i = n; i >=1; i-=2) {  
 tmp \*= i;  
 }  
 return tmp;  
 }  
}  
  
public class Task4 {  
 public static void main(String[] args) {  
 System.*out*.println(Factorial.*doubleFactorial*(5)); // = 15  
 }  
}

**Задание 5**

Напишите программу со статическим методом, которым вычисляется сумма квадратов натуральных чисел 1^2 + 2^2 + 3^2 + ... + n^2. Число п передается аргументом методу. Для проверки результата можно использовать формулу 1^2 + 2^2 +3^2+…+n^2= n(n+l) (2n + 1)/6

**Результат**

class Sum {  
 static public long quadSum(long n) {  
 long tmp = 0;  
 for (long i = 1; i <= n; i++) {  
 tmp += i \* i;  
 }  
 return tmp;  
 }  
}  
  
public class Task5 {  
 public static void main(String[] args) {  
 var n = 3;  
 var result = Sum.*quadSum*(n);  
 System.*out*.println("result: " + result);  
  
 var check = n\*(n + 1)\*(2 \* n + 1) / 6;  
 System.*out*.println("real answer: " + check);  
  
  
 }  
}

**Задание 6**

Напишите программу со статическим методом, которому аргументом передается целочисленный массив и целое число. Результатом метод возвращает ссылку на новый массив, который получается из исходного массива (переданного первым аргументом методу), если в нем взять несколько начальных элементов. Количество элементов, которые

нужно взять из исходного массива, определяются вторым аргументом метода. Если второй аргумент метода больше длины массива, переданного первым аргументом, то методом создается копия исходного массива и возвращается ссылка на эту копию.

**Результат**

class ArraySyntax {  
 public static int[] sub(int[] array, int n) {  
 if (n >= array.length) return array.clone();  
 var tmp = new int[n];  
 for (int i = 0; i < n; i++) {  
 tmp[i] = array[i];  
 }  
 return tmp;  
 }  
}  
  
public class Task6 {  
 public static void main(String[] args) {  
 var someArray = new int[] {1,2,3,4};  
  
 System.*out*.println("array: "+ Arrays.*toString*(someArray));  
 System.*out*.println("2 symbols of array: "+ Arrays.*toString*(ArraySyntax.*sub*(someArray, 2)));  
 System.*out*.println("20 symbols of array: "+ Arrays.*toString*(ArraySyntax.*sub*(someArray, 20)));  
  
 }  
}

**Задание 7**

Напишите программу со статическим методом, аргументом которому передастся

символьный массив, а результатом возвращается ссылка на целочисленным массив,

состоящий из кодов символов из массива- аргумента.

**Результат**

class Converter {  
 public static int[] convert(char[] chars) {  
 var tmp = new int[chars.length];  
 for (int i = 0; i < chars.length; i++) {  
 tmp[i] = chars[i];  
 }  
 return tmp;  
 }  
}  
  
public class Task7 {  
 public static void main(String[] args) {  
 System.*out*.println(Arrays.*toString*(Converter.*convert*(new char[]{'a', 'b', 'c'})));  
 }  
}

**Задание 8**

Напишите программу со статическим методом, аргументом которому передается целочисленный массив, а результатом возвращается среднее значение для элементов массива (сумма значений элементов, деленная на количество элементов в массиве).**Результат**

class Averager {  
 static public float average(int[] ints) {  
 float tmp = 0;  
 for (int anInt : ints) {  
 tmp += anInt;  
 }  
 return tmp / ints.length;  
 }  
}  
  
public class Task8 {  
 public static void main(String[] args) {  
 System.*out*.println(Averager.*average*(new int[]{1, 2, 3, 4, 5}));  
 }  
}

**Задание 9**

Напишите программу со статическим методом, аргументом которому передается одномерный символьный массив. В результате вызова метода элементы массива попарно меняются местами: первый — с последним, второй — с предпоследним и так далее.

**Результат**

class Shuffler {  
 static public char[] shuffle(char[] chars) {  
 var tmp = new char[chars.length];  
 for (int i = 0; i < chars.length; i++) {  
 tmp[i] = chars[chars.length - i - 1];  
 }  
 return tmp;  
 }  
}  
  
public class Task9 {  
 public static void main(String[] args) {  
 System.*out*.println(Shuffler.*shuffle*(new char[]{'a', 'b', 'c', 'd', 'e'}));  
 }  
}

**Задание 10**

Напишите программу со статическим методом, аргументом которому передается произвольное количество целочисленных аргументов. Результатом метод возвращает массив из двух элементов: это значения наибольшего и наименьшего значений среди аргументов, переданных методу.

**Результат**

class Stats {  
 static public int[] stats(int[] ints) {  
 var max = Integer.*MIN\_VALUE*;  
 var min = Integer.*MAX\_VALUE*;  
 for (int i = 0; i < ints.length; i++) {  
 max = Math.*max*(max, ints[i]);  
 min = Math.*min*(min, ints[i]);  
  
 }  
 return new int[]{max, min};  
 }  
}  
  
public class Task10 {  
 public static void main(String[] args) {  
 System.*out*.println(Arrays.*toString*(Stats.*stats*(new int[]{1, 2, 3, 4, 5})));  
 }  
}

**Вывод**

Во время выполнения работы были отработаны навыки работы с классами и конструкторами классов.