

# Практика 7

1. Добавить одну операцию по преобразованию типа, чтобы на экран вывелось 9

```
public class Solution {  
    public static void main(String[] args) {  
        short number = 9;  
        char zero = '0';  
        int nine = (zero + number);  
        System.out.println(nine);  
    }  
}
```

2. Расставьте правильно **операторы приведения типа**, чтобы получился ответ: c = 256

```
public class Solution {  
    public static void main(String[] args) {  
        int a = (byte) 44;  
        int b = (byte) 300;  
        short c = (byte) (b - a);  
        System.out.println(c);  
    }  
}
```

3. Добавить одну операцию по преобразованию типа, чтобы получался ответ: b = 0

```
public class Solution {  
    public static void main(String[] args) {  
        float f = (float) 128.50;  
        int i = (int) f;  
        int b = (int) (i + f);  
        System.out.println(b);  
    }  
}
```

4. Добавить одну **операцию по преобразованию типа**, чтобы получался ответ:  $d = 2.941$

Пример вывода: 2.9411764705882355

```
public class Solution {  
    public static void main(String[] args) {  
        int a = 50;  
        int b = 17;  
        double d = a / b;  
        System.out.println(d);  
    }  
}
```

5. Добавить одну **операцию по преобразованию типа**, чтобы получался ответ:  $d = 5.5$

```
public class Solution {  
    public static void main(String[] args) {  
        int a = 5;  
        int b = 4;  
        int c = 3;  
        int e = 2;  
        double d = a + b / c / e;  
        System.out.println(d);  
    }  
}
```

6. Добавить одну **операцию по преобразованию типа**, чтобы получался ответ:  $d = 1.0$

```
public class Solution {  
    public static void main(String[] args) {  
        int a = 257;  
        int b = 4;  
        int c = 3;  
        int e = 2;  
        double d = a + b / c / e;  
  
        System.out.println(d);  
    }  
}
```

7. Вывести на экран надпись "Я не хочу есть блины, я хочу оладушки" 25 раз по образцу.

Образец:

Я не хочу есть блины, я хочу оладушки  
не хочу есть блины, я хочу оладушки  
не хочу есть блины, я хочу оладушки  
е хочу есть блины, я хочу оладушки

```
public class Solution {  
    public static void main(String[] args) {  
        String s = "Я не хочу есть блины, я хочу оладушки";  
    }  
}
```

8. Ввести с клавиатуры 10 строчек и подсчитать в них количество различных букв (для 33 маленьких букв алфавита). Результат вывести на экран в алфавитном порядке.

Пример вывода:

а 5  
б 8  
в 3  
г 7  
д 0  
...  
я 9

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        List<Character> alphabet = Arrays.asList(  
            'а', 'б', 'в', 'г', 'д', 'е', 'ё', 'ж',  
            'з', 'и', 'й', 'к', 'л', 'м', 'н', 'о',  
            'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц',  
            'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я');  
        // напишите тут ваш код  
    }  
}
```

9. Создать **массив**, элементами которого будут **списки строк**. Заполнить массив **любыми данными** и вывести их на экран.

```
public class Solution {
    public static void main(String[] args) {
        ArrayList<String>[] arrayOfStringList = createList();
        printList(arrayOfStringList);
    }

    public static ArrayList<String>[] createList() {
        //напишите тут ваш код
        return null;
    }

    public static void printList(ArrayList<String>[] arrayOfStringList) {
        for (ArrayList<String> list : arrayOfStringList) {
            for (String s : list) {
                System.out.println(s);
            }
        }
    }
}
```

10. Расставьте как можно меньше модификаторов **static** так, чтобы пример скомпилировался.

```
public class Solution {
    public int A = 5;
    public int B = 5;
    public int C = 5;
    public static int D = 5;

    public static void main(String[] args) {
        Solution solution = new Solution();
        solution.A = 5;
        solution.B = 5 * B;
        solution.C = 5 * C * D;
        Solution.D = 5 * D * C;
        Solution.D = 5;
    }

    public int getA() {
        return A;
    }
}
```

11. Напишите класс **Human** с **6** полями. Придумайте и реализуйте **10** различных конструкторов для него. Каждый конструктор должен иметь смысл.

```
public class Solution {  
    public static void main(String[] args) {  
    }  
  
    public static class Human {  
        // Напишите тут ваши переменные и конструкторы  
    }  
}
```

12. Ввести с клавиатуры в список **20** слов. Нужно подсчитать сколько раз каждое слово встречается в списке. Результат нужно представить в виде словаря **Map<String, Integer>**, где **первый параметр - уникальное слово**, а **второй - ЧИСЛО**, сколько раз данное слово встречалось в списке. Вывести содержимое словаря на экран. В тестах регистр (большая/маленькая буква) влияет на результат.

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        ArrayList<String> words = new ArrayList<String>();  
        for (int i = 0; i < 20; i++) {  
            words.add(reader.readLine());  
        }  
  
        Map<String, Integer> map = countWords(words);  
        for (Map.Entry<String, Integer> pair : map.entrySet()) {  
            System.out.println(pair.getKey() + " " + pair.getValue());  
        }  
    }  
  
    public static Map<String, Integer> countWords(ArrayList<String> list) {  
        HashMap<String, Integer> result = new HashMap<String, Integer>();  
        //напишите тут ваш код  
        return result;  
    }  
}
```

13. Создать список целых чисел.

Ввести с клавиатуры **20** целых чисел.

Создать метод по безопасному извлечению чисел из списка: **int safeGetElement(ArrayList<Integer> list, int index, int defaultValue)**

Метод должен возвращать элемент списка (**list**) по его индексу (**index**).

Если в процессе получения элемента возникло исключение, его нужно перехватить, и метод должен вернуть **defaultValue**.

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        ArrayList<Integer> list = new ArrayList<Integer>();  
        for (int i = 0; i < 20; i++) {  
            int x = Integer.parseInt(reader.readLine());  
            list.add(x);  
        }  
        System.out.println(safeGetElement(list, 5, 1));  
        System.out.println(safeGetElement(list, 20, 7));  
        System.out.println(safeGetElement(list, -5, 9));  
    }  
  
    public static int safeGetElement(ArrayList<Integer> list, int index, int defaultValue) {  
        //напишите тут ваш код  
    }  
}
```

14. Задача: ввести с клавиатуры **30** чисел. Вывести **10-е** и **11-е минимальные числа**. Пояснение: Самое минимальное число - 1-е минимальное. Следующее минимальное после него - 2-е минимальное

Пример: 1 6 5 7 1 15 63 88

Первое минимальное - 1

Второе минимальное - 1

Третье минимальное - 5

Четвертое минимальное - 6

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        int[] array = new int[30];  
        for (int i = 0; i < 30; i++) {  
            array[i] = Integer.parseInt(reader.readLine());  
        }  
        sort(array);  
        System.out.println(array[9]);  
        System.out.println(array[10]);  
    }  
  
    public static void sort(int[] array) {  
        //напишите тут ваш код  
    }  
}
```



15. Задача: Программа демонстрирует работу **HashMap**: вводит с клавиатуры набор пар (**номер и строку**), помещает их в **HashMap** и выводит на экран содержимое **HashMap**.

```
public class Solution {  
    HashMap<int, String> map;  
    static Integer index;  
    static String name;  
  
    public Solution() {  
        this.map = new HashMap<int, String>();  
        map.put(index, name);  
    }  
  
    public static void main(String[] args) throws IOException {  
        Solution solution = new Solution();  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        for (int i = 0; i < 10; i++) {  
            int index = Integer.parseInt(reader.readLine());  
            String name = reader.readLine();  
            solution.map.put(index, name);  
        }  
        for (Map.Entry<int, String> pair : solution.map.entrySet()) {  
            index = pair.getKey();  
            name = pair.getValue();  
            System.out.println("Id=" + index + " Name=" + name);  
        }  
    }  
}
```

16. Задача: Программа вводит с клавиатуры пару (**число и строку**) и выводит их на экран.

Новая задача: Программа вводит с клавиатуры пары (**число и строку**), сохраняет их в `HashMap`. Пустая строка - конец ввода данных. Числа могут повторяться. Строки всегда уникальны. Введенные данные не должны потеряться! Затем программа выводит содержание `HashMap` на экран.

Каждую пару с новой строки. Пример ввода:

1

Мама

2

Папа

```
public class Solution {  
    public static void main(String[] args) throws IOException {  
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));  
        int id = Integer.parseInt(reader.readLine());  
        String name = reader.readLine();  
        System.out.println("Id=" + id + " Name=" + name);  
    }  
}
```