

Практика 4

1. В каждом классе **Cat** и **Dog** написать метод **finalize**, который выводит на экран текст о том, что такой-то объект уничтожен.

```
public class Cat {  
    public static void main(String[] args) {  
  
    }  
}
```

//напишите тут ваш код

}

```
class Dog {  
    //напишите тут ваш код  
}
```

2. В конструкторе класса **Cat [public Cat()]** увеличивать счётчик котов (статическую переменную этого же класса **catCount**) на **1**. В методе **finalize** уменьшать на **1**.

```
public class Cat {  
    public static int catCount = 0;
```

//напишите тут ваш код

```
    public static void main(String[] args) {
```

```
    }
```

```
}
```

3. Создать в цикле по **5 000 000** объектов **Cat** и **Dog**. (Java-машина должна начать уничтожать неиспользуемые, и метод `finalize` хоть раз да вызовется).

```
public class Solution {  
    public static void main(String[] args) {  
        // напишите тут ваш код  
    }  
}
```

```
class Cat {  
    @Override  
    protected void finalize() throws Throwable {  
        super.finalize();  
        System.out.println("A Cat was destroyed");  
    }  
}
```

```
class Dog {  
    @Override  
    protected void finalize() throws Throwable {  
        super.finalize();  
        System.out.println("A Dog was destroyed");  
    }  
}
```

4. Ввести с клавиатуры **число**. Определить, сколько в введенном числе **четных цифр**, а сколько **нечетных**. Если число делится без остатка на **2** (т. е. остаток равен нулю), значит оно **четное**. Увеличиваете на **1** счетчик четных цифр (статическая переменная `even`). Иначе число **нечетное**, увеличиваете счетчик **нечетных цифр** (статическая переменная `odd`). Вывести на экран сообщение: "**Even: a Odd: b**", где `a` - количество четных цифр, `b` - количество нечетных цифр. Пример для числа 4445: Even: 3 Odd: 1

```
public class Solution {  
  
    public static int even;  
    public static int odd;  
    public static void main(String[] args) throws IOException {  
        //напишите тут ваш код  
    }  
}
```

5. Создать статическую переменную `int catCount` в классе **Cat**. Создайте конструктор `[public Cat()]`, в котором увеличивай данную переменную на **1**.

```
public class Cat {  
    //напишите тут ваш код  
  
    public static void main(String[] args) {  
  
    }  
}
```

6. Сделать класс **ConsoleReader**, у которого будут **4** статических метода:

String readString() - читает с клавиатуры строку

int readInt() - читает с клавиатуры число

double readDouble() - читает с клавиатуры дробное число

boolean readBoolean() - читает с клавиатуры строку "**true**" или "**false**" и возвращает соответствующую логическую переменную true или false

Внимание: создавайте переменную для чтения данных с консоли (**BufferedReader** или **Scanner**) внутри каждого метода.

```
public class ConsoleReader {  
    public static String readString() throws Exception {  
        //напишите тут ваш код  
    }  
  
    public static int readInt() throws Exception {  
        //напишите тут ваш код  
    }  
  
    public static double readDouble() throws Exception {  
        //напишите тут ваш код  
    }  
  
    public static boolean readBoolean() throws Exception {  
        //напишите тут ваш код  
    }  
  
    public static void main(String[] args) throws Exception {  
  
    }  
}
```

7. Программа должна считывать введенные пользователем **вес** в **килограммах** и **рост** в метрах и выводить на экран сообщение о **индексе массы тела**. Реализуйте статический метод класса `Body`. Метод должен **определить индекс массы тела**, и вывести на экран сообщение: "**Недовес: меньше чем 18.5**" - если индекс массы тела меньше **18.5** (не включая), "**Нормальный: между 18.5 и 25**" - если индекс массы тела между **18.5** и **25** (не включая), "**Избыточный вес: между 25 и 30**" - если индекс массы тела между **25** и **30** (не включая), "**Ожирение: 30 или больше**" - если индекс массы тела **30** или больше. Подсказка: Индекс массы тела = вес в кг / (рост в метрах * рост в метрах) Пример вывода для **68.4** и **1.77**: Нормальный: между 18.5 и 25

```
public class Solution {
```

```
    public static void main(String[] args) throws IOException {
        BufferedReader bis = new BufferedReader(new InputStreamReader(System.in));
        double weight = Double.parseDouble(bis.readLine());
        double height = Double.parseDouble(bis.readLine());

        Body.calculateMassIndex(weight, height);
    }
```

```
    public static class Body {
        public static void calculateMassIndex(double weight, double height) {
            // напишите тут ваш код
        }
    }
}
```

8. Сделать класс **Calculator**, у которого будут **5** статических методов:

int plus(int a, int b) - возвращает сумму чисел **a** и **b**

int minus(int a, int b) - возвращает разницу чисел **a** и **b**

int multiply(int a, int b) - возвращает результат умножения числа **a** на число **b**

double division(int a, int b) - возвращает результат деления числа **a** на число **b**

double percent(int a, int b) - возвращает **b** процентов от числа **a** (например, `percent(4, 50)` должен вернуть 50% от 4)

```
public class Calculator {  
    public static int plus(int a, int b) {  
        //напишите тут ваш код  
        return 0;  
    }  
  
    public static int minus(int a, int b) {  
        //напишите тут ваш код  
        return 0;  
    }  
  
    public static int multiply(int a, int b) {  
        //напишите тут ваш код  
        return 0;  
    }  
  
    public static double division(int a, int b) {  
        //напишите тут ваш код  
        return 0;  
    }  
  
    public static double percent(int a, int b) {  
        //напишите тут ваш код  
        return 0;  
    }  
  
    public static void main(String[] args) {  
  
    }  
}
```

9. В классе **Cat** создайте статическую переменную `public int catCount`. Создайте конструктор `[public Cat()]`. Пусть при каждом создании кота (нового объекта **Cat**) статическая переменная `catCount` увеличивается на **1**. Создать **10** объектов **Cat** и вывести значение переменной `catCount` на экран.

```
public class Solution {  
    public static void main(String[] args) {  
        // Создайте 10 котов  
  
        // Выведите значение переменной catCount  
    }  
  
    public static class Cat {  
        // Создайте статическую переменную catCount  
  
        // Создайте конструктор  
    }  
}
```

10. 1. В классе **Cat** добавьте публичную статическую переменную `cats` (`ArrayList<Cat>`).
2. Пусть при каждом создании кота (нового объекта **Cat**) в переменную `cats` добавляется этот новый кот. Создать **10** объектов **Cat**.
3. Метод `printCats` должен выводить всех котов на экран. Нужно использовать переменную `cats`.

```
public class Cat {  
    //напишите тут ваш код  
  
    public Cat() {  
    }  
  
    public static void main(String[] args) {  
        //напишите тут ваш код  
        printCats();  
    }  
  
    public static void printCats() {  
        //напишите тут ваш код  
    }  
}
```


11. Расставьте минимальное количество (4шт.) **static-ов**, чтобы код начал работать, и программа успешно завершилась.

```
public class Solution {  
    public int step;  
  
    public static void main(String[] args) {  
        method1();  
    }  
  
    public void method1() {  
        method2();  
    }  
  
    public void method2() {  
        new Solution().method3();  
    }  
  
    public void method3() {  
        method4();  
    }  
  
    public void method4() {  
        step++;  
        for (StackTraceElement element : Thread.currentThread().getStackTrace())  
            System.out.println(element);  
        if (step > 1)  
            return;  
        main(null);  
    }  
}
```

12. 1. В классе Solution создать public static класс Idea.
2. В классе **Idea** создать метод `public String getDescription()`, который будет возвращать любую строку.
3. В классе Solution создайте статический метод `public void printIdea(Idea idea)`, который будет выводить на экран описание идеи - это то, что возвращает метод `getDescription()`.

```
public class Solution {  
    public static void main(String[] args) {  
        printIdea(new Idea());  
    }  
}
```

//напишите тут ваш код

13. Добавьте **3** public статических переменных: `String Solution.name`, `String Cat.name`, `String Dog.name`.

```
public class Solution {  
    public static class Cat {  
    }  
  
    public static class Dog {  
    }  
  
    public static void main(String[] args) {  
    }  
}
```

14. **Задача:** Программа вводит **два числа** с клавиатуры и выводит их максимум в виде **"The max is 25"**.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

```
public class Solution {
    public static int max = 100;

    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        String max = "The max is ";
        int a = Integer.parseInt(reader.readLine());
        int b = Integer.parseInt(reader.readLine());
        max = a > b ? a : b;

        System.out.println(max + max);
    }
}
```

15. Задача: У каждой кошки есть **имя** и **кошка-мама**. Создать **класс**, который бы описывал данную ситуацию. Создать **два объекта**: **кошку-дочь** и **кошку-маму**. Вывести их на экран.

Новая задача: У каждой кошки есть **имя**, **кот-папа** и **кошка-мама**. Изменить класс **Cat** так, чтобы он мог описать данную ситуацию. Создать **6 объектов**: **дедушку** (папин папа), **бабушку** (мамина мама), **папу**, **маму**, **сына**, **дочь**. Вывести их всех на экран в порядке: **дедушка, бабушка, папа, мама, сын, дочь**.

Пример ввода:

дедушка Вася
бабушка Мурка
папа Котофей
мама Василиса
сын Мурчик
дочь Пушинка

Пример вывода:

The cat's name is дедушка Вася, no mother, no father

The cat's name is бабушка Мурка, no mother, no father

The cat's name is папа Котофей, no mother, father is дедушка Вася

The cat's name is мама Василиса, mother is бабушка Мурка, no father

The cat's name is сын Мурчик, mother is мама Василиса, father is папа Котофей

The cat's name is дочь Пушинка, mother is мама Василиса, father is папа Котофей

● **см. следующий слайд**

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

        String motherName = reader.readLine();
        Cat catMother = new Cat(motherName);

        String daughterName = reader.readLine();
        Cat catDaughter = new Cat(daughterName, catMother);

        System.out.println(catMother);
        System.out.println(catDaughter);
    }

    public static class Cat {
        private String name;
        private Cat parent;

        Cat(String name) {
            this.name = name;
        }

        Cat(String name, Cat parent) {
            this.name = name;
            this.parent = parent;
        }

        @Override
        public String toString() {
            if (parent == null)
                return "The cat's name is " + name + ", no mother ";
            else
                return "The cat's name is " + name + ", mother is " + parent.name;
        }
    }
}

```

16. 1. В методе `initializeArray()`:

1.1. Создайте массив на **20** чисел

1.2. Считайте с консоли **20** чисел и заполните ими массив

2. Метод `max(int[] array)` должен находить максимальное **число из элементов массива**

```
import java.io.IOException;
```

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        int[] array = initializeArray();  
        int max = max(array);  
        System.out.println(max);  
    }
```

```
    public static int[] initializeArray() throws IOException {  
        // создайте и заполните массив  
        return null;  
    }
```

```
    public static int max(int[] array) {  
        // найдите максимальное значение  
        return 0;  
    }  
}
```

17. 1. Создать массив на **10** строк.

2. Ввести с клавиатуры **8** строк и сохранить их в массив.

3. Вывести содержимое всего массива (**10** элементов) на экран в обратном порядке.
Каждый элемент - с новой строки.

18. 1. Создать массив на **10** строк.
2. Создать массив на **10** чисел.
3. Ввести с клавиатуры **10** строк, заполнить ими массив строк.
4. В **каждую ячейку массива** чисел записать длину **строки из массива строк**, **индекс/номер** ячейки которой совпадает с текущим индексом из массива чисел.
Вывести содержимое массива чисел на экран, каждое значение выводить с новой строки.
19. 1. Создать массив на **10** чисел.
2. Ввести с клавиатуры **10** чисел и записать их в массив.
3. Вывести на экран элементы массива в обратном порядке, каждое значение выводить с новой строки.
20. 1. Создать массив на **20** чисел.
2. Ввести в него значения с клавиатуры.
3. Создать два массива на **10** чисел каждый.
4. Скопировать **большой массив в два маленьких**: **половину чисел** в первый маленький, **вторую половину** во второй маленький.
5. Вывести второй маленький массив на экран, каждое значение выводить с новой строки.
21. 1. Создать массив на **15** целых чисел.
2. Ввести в него значения с клавиатуры.
3. Пускай **индекс** элемента массива является **номером дома**, а **значение** - **числом жителей**, проживающих в доме.
Дома с **нечетными номерами** расположены на одной стороне улицы, с **четными** - на другой.
Выяснить, на какой стороне улицы проживает больше жителей.
4. Вывести на экран сообщение: "В домах с нечетными номерами проживает больше жителей." или "В домах с четными номерами проживает больше жителей."
Примечание: дом с порядковым номером 0 считать четным.

- Далее будут такие обозначения:
Список (анг. List), чаще всего под этим словом подразумевается **ArrayList**.
Строка (анг. String), под этим подразумевается тип **String**.
Создать список строк чаще всего означает: `ArrayList<String> list = new ArrayList<String>();`

22. 1. Создайте список строк.
2. Добавьте в него **5** различных строк.
3. Выведите его размер на экран.
4. Используя **цикл** выведите его содержимое на экран, каждое значение с новой строки.

23. 1. Создайте список строк.
2. Считайте с клавиатуры **5 строк** и добавьте в список.
3. Используя **цикл**, найдите **самую длинную строку** в списке.
4. Выведите найденную строку на экран. Если таких строк несколько, выведите **каждую с новой строки**.

24. Задача: Написать программу, которая вводит с клавиатуры **20 чисел** и выводит их в убывающем порядке.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
```

```
public class Solution {
    public static void main(String[] args) throws Exception {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        int[] array = new int[20];
        for (int i = 0; i < 20; i++) {
            array[i] = Integer.parseInt(reader.readLine());
        }

        sort(array);

        for (int x : array) {
            System.out.println(x);
        }
    }

    public static void sort(int[] array) {
        //напишите тут ваш код
    }
}
```

25. 1. Создайте список строк.
2. Считайте с клавиатуры **5 строк** и добавьте в список.
3. Используя **цикл**, найдите **самую короткую строку** в списке.
4. Выведите найденную строку на экран.
5. Если таких строк несколько, выведите **каждую с новой строки**.
26. 1. Создайте список строк в методе **main**.
2. Добавьте в него **10 строчек** с клавиатуры, но только добавлять не в конец списка, а в начало.
3. Используя **цикл** выведите содержимое на экран, каждое значение с новой строки.
27. 1. Создайте список строк.
2. Добавьте в него **5 строк** с клавиатуры.
3. Выполните **13 раз**: удалить последнюю строку и вставить её в начало.
4. Используя **цикл** выведите содержимое результирующего списка на экран, **каждое значение с новой строки**.
28. 1. Создайте список строк.
2. Добавьте в него **10 строчек** с клавиатуры.
3. Узнайте, какая строка в списке встретится раньше: **самая короткая** или **самая длинная**.
Если таких строк **несколько**, то должны быть учтены **самые первые** из них.
4. Выведи на экран строку из **п.3**. Должна быть выведена одна строка.

29. Введите с клавиатуры **5 слов** в список строк. Удалите **3**-ий элемент списка, и выведите оставшиеся элементы в обратном порядке.
30. 1. Создайте список из слов "**мама**", "**мыла**", "**раму**".
2. После каждого слова вставьте в список строку, содержащую слово «**именно**».
3. Вывести результат на экран, каждый элемент списка с новой строки.
31. 1. Введите с клавиатуры **10 слов** в список строк.
2. Определить, является ли список **упорядоченным** по возрастанию длины строки.
3. В случае отрицательного ответа вывести на экран **индекс первого элемента**, нарушающего такую упорядоченность.

32. 1. Введите с клавиатуры **10 слов** в список строк.
2. Метод `doubleValues` должен удваивать слова по принципу:
"альфа", "бета", "гамма" -> "альфа", "альфа", "бета", "бета", "гамма",
"гамма"
3. Выведите результат на экран, каждое значение с новой строки.

```
public class Solution {  
    public static void main(String[] args) throws Exception {  
        // Считать строки с консоли и объявить ArrayList list тут  
  
        ArrayList<String> result = doubleValues(list);  
  
        // Вывести на экран result  
    }  
  
    public static ArrayList<String> doubleValues(ArrayList<String> list) {  
        //напишите тут ваш код  
        return null;  
    }  
}
```

33. 1. Создайте список слов, заполните его самостоятельно.

2. Метод **fix** должен:

2.1. удалять из списка строк все слова, содержащие букву "р"

2.2. удваивать все слова содержащие букву "л".

2.3. если слово содержит и букву "р" и букву "л", то оставить это слово без изменений.

2.4. с другими словами ничего не делать.

Пример:

роза

лира

лоза

Выходные данные:

лира

лоза

лоза

```
public class Solution {  
    public static void main(String[] args) {  
        ArrayList<String> strings = new ArrayList<String>();  
        strings.add("роза");  
        strings.add("лоза");  
        strings.add("лира");  
        strings = fix(strings);  
  
        for (String string : strings) {  
            System.out.println(string);  
        }  
    }  
  
    public static ArrayList<String> fix(ArrayList<String> strings) {  
        //напишите тут ваш код  
        return null;  
    }  
}
```

34. Ввести с клавиатуры **10 чисел** и заполнить ими список.
Вывести их **в обратном порядке**. Каждый элемент - с новой строки.
Использовать только цикл `for`.

35. Ввести с клавиатуры **2** числа **N** и **M**.
Ввести **N** строк и заполнить ими список.
Переставить **M** первых строк в конец списка.
Вывести список на экран, каждое значение с новой строки.
Примечание: запрещено создавать больше одного списка.

36. Создать массив на **20 чисел**.
Заполнить его числами с клавиатуры.
Найти **максимальное** и **минимальное** числа в массиве.
Вывести на экран максимальное и минимальное числа через пробел.

37. Создать список строк.
Ввести строки с клавиатуры и добавить их в список.
Вводить с клавиатуры строки, пока пользователь не введет строку **"end"**. Саму строку **"end"** не учитывать.
Вывести строки на экран, каждую с новой строки.

38. Написать программу, которая ведёт обратный отсчёт с **30** до **0**, а в конце выводит на экран текст **"Бум!"**.
Программа должна уменьшать число **10 раз** в секунду. Для того чтобы вставить в программу задержку, воспользуйся функцией:
`Thread.sleep(100);` //задержка на одну десятую секунды.

Пример:

30

29

...

1

0

Бум!

39. Создайте класс **Human** с полями **имя**(**String**), **пол**(**boolean**), **возраст**(**int**), **отец**(**Human**), **мать**(**Human**).
Создайте объекты и заполните их так, чтобы получилось: Два дедушки, две бабушки, отец, мать, трое детей. Вывести объекты на экран.

Примечание:

Если написать свой метод **String toString()** в классе **Human**, то именно он будет использоваться при выводе объекта на экран.

Пример вывода:

Имя: Аня, пол: женский, возраст: 21, отец: Павел, мать: Катя

Имя: Катя, пол: женский, возраст: 55

Имя: Игорь, пол: мужской, возраст: 2, отец: Михаил, мать: Аня

...

```
public class Solution {  
    public static void main(String[] args) {  
        // напишите тут ваш код  
    }  
  
    public static class Human {  
        // напишите тут ваш код  
  
        public String toString() {  
            String text = "";  
            text += "Имя: " + this.name;  
            text += ", пол: " + (this.sex ? "мужской" : "женский");  
            text += ", возраст: " + this.age;  
  
            if (this.father != null) {  
                text += ", отец: " + this.father.name;  
            }  
  
            if (this.mother != null) {  
                text += ", мать: " + this.mother.name;  
            }  
  
            return text;  
        }  
    }  
}
```

40. Задача: Программа вводит с клавиатуры **данные про котов** и выводит их на экран.

Пример:

Cat's name: Barsik, age: 6, weight: 5, tail: 22

Cat's name: Murka, age: 8, weight: 7, tail: 20

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
public class Solution {
    public final static ArrayList<Cat> CATS = new ArrayList<>();
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            String name = reader.readLine();
            if (name == null || name.isEmpty()) {
                break;
            }
            Cat cat = new Cat(name);
            CATS.add(cat);
        }
        printList();
    }
    public static void printList() {
        for (int i = 0; i < CATS.size(); i++) {
            System.out.println(CATS.get(i));
        }
    }
    public static class Cat {
        private String name;
        private int age;
        private int weight;
        private int tailLength;
        Cat(String name, int age, int weight, int tailLength) {
            this.name = name;
            this.age = age;
            this.weight = weight;
            this.tailLength = tailLength;
        }
        @Override
        public String toString() {
            return "Cat's name: " + name + ", age: " + age + ", weight: " + weight + ", tail: " + tailLength;
        }
    }
}
```


41. **Задача:** Программа вводит **строки**, пока пользователь не введёт **пустую строку** (нажав **enter**). Потом она конвертирует строки в верхний регистр (**Мама** превращается в **MAMA**) и выводит их на экран.

Новая задача: Программа вводит строки, пока пользователь не введёт пустую строку (нажав **enter**).

Потом программа строит **новый список**. Если в строке чётное число букв, строка удваивается, если нечётное - утраивается.

Программа выводит содержимое нового списка на экран.

Пример ввода:

Кот

Коты

Я

Пример вывода:

Кот Кот Кот

Коты Коты

Я Я Я

```
import java.io.BufferedReader;
```

```
import java.io.IOException;
```

```
import java.io.InputStreamReader;
```

```
import java.util.ArrayList;
```

```
public class Solution {
```

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

```
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
```

```
        ArrayList<String> strings = new ArrayList<String>();
```

```
        while (true) {
```

```
            String string = reader.readLine();
```

```
            if (string == null || string.isEmpty()) break;
```

```
            strings.add(string);
```

```
        }
```

```
        ArrayList<String> resultStrings = new ArrayList<String>();
```

```
        for (int i = 0; i < strings.size(); i++) {
```

```
            String string = strings.get(i);
```

```
            resultStrings.add(string.toUpperCase());
```

```
        }
```

```
        for (int i = 0; i < resultStrings.size(); i++) {
```

```
            System.out.println(resultStrings.get(i));
```

```
        }
```

```
    }
```

```
}
```