

Пример 1. Вывод произвольного числа целочисленных аргументов

```
public class varargs{
    public static void printArgument(int... k) {
        System.out.println("Количество аргументов: " +
k.length);
        for(int i = 0; i < k.length; i++) {
            System.out.println(k[i]);
        }
    }
    public static void main(String[] args) {
        printArgument(1,2,3,4,5);
    }
}
```

Пример 2. Вывод произвольного числа строковых аргументов.
Перегруженная функция printArgument

```
public class varargs{
    public static void printArgument(int... k) {
        System.out.println("Количество аргументов: " +
k.length);
        for(int i = 0; i < k.length; i++) {
            System.out.println(k[i]);
        }
    }
    public static void printArgument(String... k) {
        System.out.println("Количество аргументов: " +
k.length);
        for(int i = 0; i < k.length; i++) {
            System.out.println(k[i]);
        }
    }
    public static void main(String[] args) {
        printArgument("a","bb","ccc");
    }
}
```

Пример 3. Метод, принимающий аргументы любого типа

```
public class varargs{
    static void ShowInfo(Object ... objects) {
        // вывод информации об аргументах, переданных в метод
        for (Object o : objects)
            System.out.println(o.toString());
        System.out.println("Кол-во аргументов: " + objects.length);
    }

    public static void main(String[] args) {
        ShowInfo(1, "aa", 'd', 3.1415);
    }
}
```

Пример 4. Аргументы – массивы

```
import java.util.Arrays;

public class varargs {
    static void ShowInfo(Object... arr) {
        System.out.println(Arrays.deepToString(arr));
        System.out.println("Кол-во аргументов: " +
arr.length);
    }

    public static void main(String[] args) {
        Integer[] a1 = {1, 2, 3, 4, 5};
        Integer[] a2 = {100, 200};
        Integer[] a3 = {3, 7, 21,};
        ShowInfo(a1, a2, a3);
    }
}
```

Пример 5. Аргументы – экземпляры класса

```
class Product {
    String name;
    float price;

    public Product(String name, float price) {
        this.name = name;
        this.price = price;
    }

    public String getName() {
        return name;
    }

    public float getPrice() {
        return price;
    }

    public static void ShowInfo(Product p) {
        System.out.println(p.getName()+" "+p.getPrice());
    }

    static Product GetMaxPrice(Product ... PP) {
        Product p=null;
        if (PP.length==0) {
            return p;
        }
        p = PP[0];
        for (int i=1; i<PP.length; i++)
            if (p.price<PP[i].price)
                p = PP[i];
        return p;
    }
}

public class Main {
    public static void main(String[] args) {

        Product p1= new Product("pen",100);
        Product p2= new Product("pencil",120);
        Product p3= new Product("notebook",150);

        Product answer = Product.GetMaxPrice(p1,p2);
        Product.ShowInfo(answer);

        answer = Product.GetMaxPrice(p1,p2,p3);
        Product.ShowInfo(answer);
    }
}
```

Задания

1. Написать метод, принимающий произвольное количество аргументов типа `int`, который выводит элементы, их количество их сумму.
2. Переопределить метод из первого задания для типов `char` и `String`.
3. Переписать метод (возвращает строку) из первого задания для произвольного количества `Object` (исключая массивы). Сумма объектов – строка.
4. Написать методы, возвращающие экземпляр класса `Product` (поля: имя, цена, количество)
 - a. с минимальной ценой товара
 - b. с самым длинным названием
 - c. с количеством, наиболее близким к среднему количеству из всех товаров.
5. Написать метод, получающий в качестве параметров два целочисленных аргумента m – количество строк, n – количество столбцов матрицы, и произвольное количество целочисленных аргументов – элементы матрицы. Метод возвращает двумерный массив размера $n \times m$.