

The image shows a screenshot of an IDE with two panels. The top panel displays the source code of a Java program named `Main.java`. The code defines a package `com.company` and a class `UsingStatic`. Inside the class, there are static fields `a` and `b`, a static method `meth`, and a static block. The `main` method calls `meth` with the argument `123`. The bottom panel shows the execution output for the `UsingStatic` class. The output displays the static block's effect first, then the values of `x`, `a`, and `b` after the `meth` method is executed. The process finished with exit code 0.

```
1 package com.company;
2
3 class UsingStatic{
4     // Статические поля:
5     static int a=3;
6     static int b;
7     // Статический метод:
8     static void meth(int x){
9         System.out.println("x="+x);
10        System.out.println("a="+a);
11        System.out.println("b="+b);}
12    // Статический блок:
13    static{
14        System.out.println("Статический блок:");
15        b=a*4;}
16    // Вызов статического метода:
17    public static void main(String args[]){
18        meth(123);
19    }
20 }
```

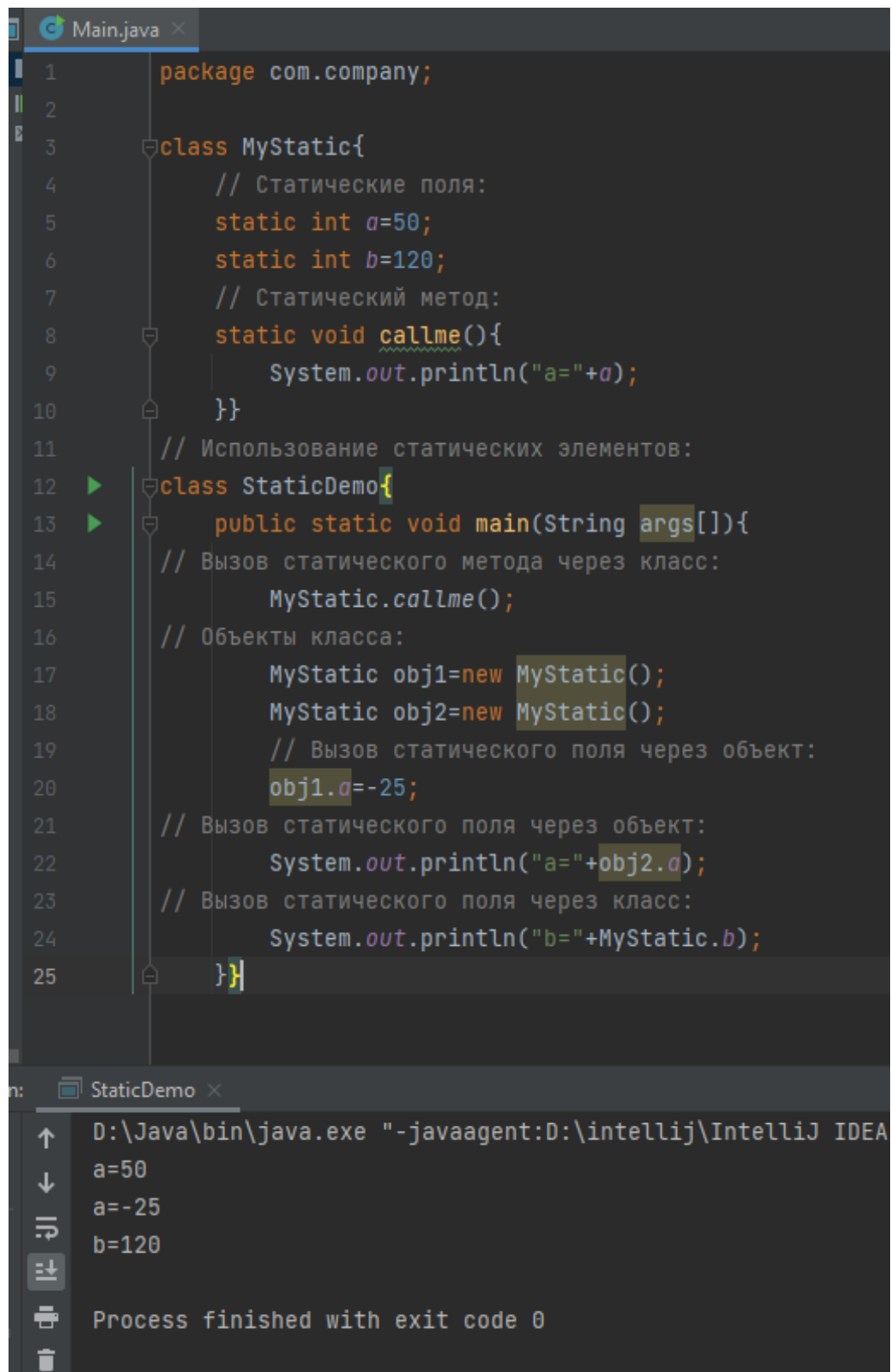
UsingStatic ×

↑ D:\Java\bin\java.exe "-javaagent:D:\intellij\IntelliJ IDEA
Статический блок:
x=123
a=3
b=12

↓

Process finished with exit code 0

Листинг 4.3



The screenshot shows an IDE with two windows. The top window, titled 'Main.java', contains the following Java code:

```
1 package com.company;
2
3 class MyStatic{
4     // Статические поля:
5     static int a=50;
6     static int b=120;
7     // Статический метод:
8     static void callme(){
9         System.out.println("a="+a);
10    }
11    // Использование статических элементов:
12    class StaticDemo{
13        public static void main(String args[]){
14            // Вызов статического метода через класс:
15            MyStatic.callme();
16            // Объекты класса:
17            MyStatic obj1=new MyStatic();
18            MyStatic obj2=new MyStatic();
19            // Вызов статического поля через объект:
20            obj1.a=-25;
21            // Вызов статического поля через объект:
22            System.out.println("a="+obj2.a);
23            // Вызов статического поля через класс:
24            System.out.println("b="+MyStatic.b);
25        }
26    }
```

The bottom window, titled 'StaticDemo', shows the execution output:

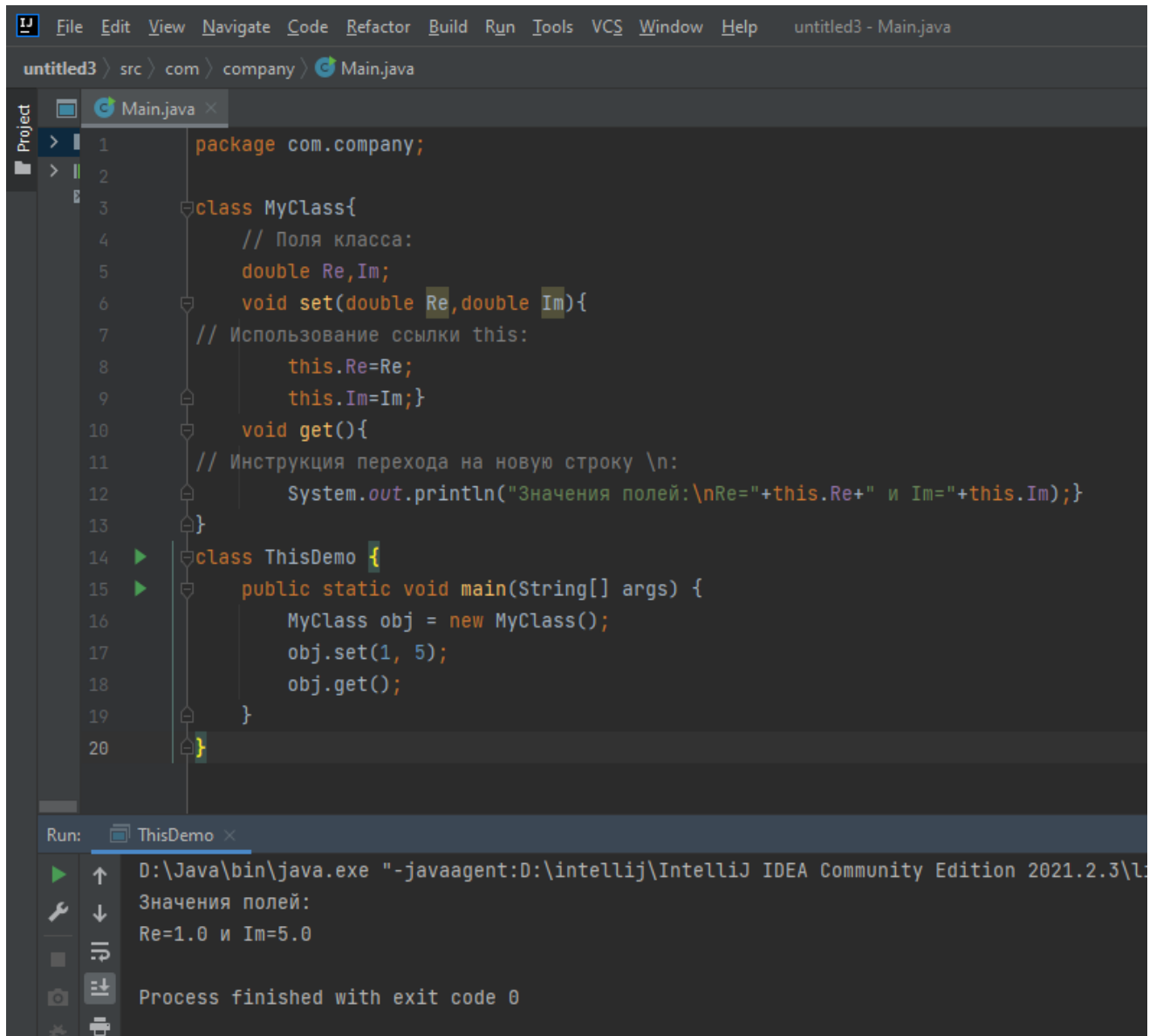
```
D:\Java\bin\java.exe "-javaagent:D:\intellij\IntelliJ IDEA
a=50
a=-25
b=120
Process finished with exit code 0
```

Листинг 4.4

```
Main.java x
1 package com.company;
2
3 class MyClass{
4     // Закрытые поля:
5     private int a;
6     private int b;
7     // Закрытый метод:
8     private void showab(){
9         // Обращение к закрытым полям в классе:
10        System.out.println("a="+a);
11        System.out.println("b="+b);}
12    // Открытый метод:
13    public void setab(int x,int y){
14        // Обращение к закрытым полям в классе:
15        a=x;
16        b=y;
17        System.out.println("Присвоены значения полям!");}
18    // Открытый метод:
19    void getab(){
20        System.out.println("Проверка значений полей:");
21        // Обращение к закрытому методу в классе:
22        showab();}
23 }
24 class PrivateDemo{
25     public static void main(String[] args){
26         // Создание объекта:
27         MyClass obj=new MyClass();
28         // Вызов открытых методов:
29         obj.setab( x: 3, y: 5);
30         obj.getab();
31     }}
```

```
PrivateDemo x
↑ D:\Java\bin\java.exe "-javaagent:D:\intellij\IntelliJ IDEA Communi
↓ Присвоены значения полям!
: Проверка значений полей:
: a=3
: b=5
:
: Process finished with exit code 0
```

Листинг 4.5

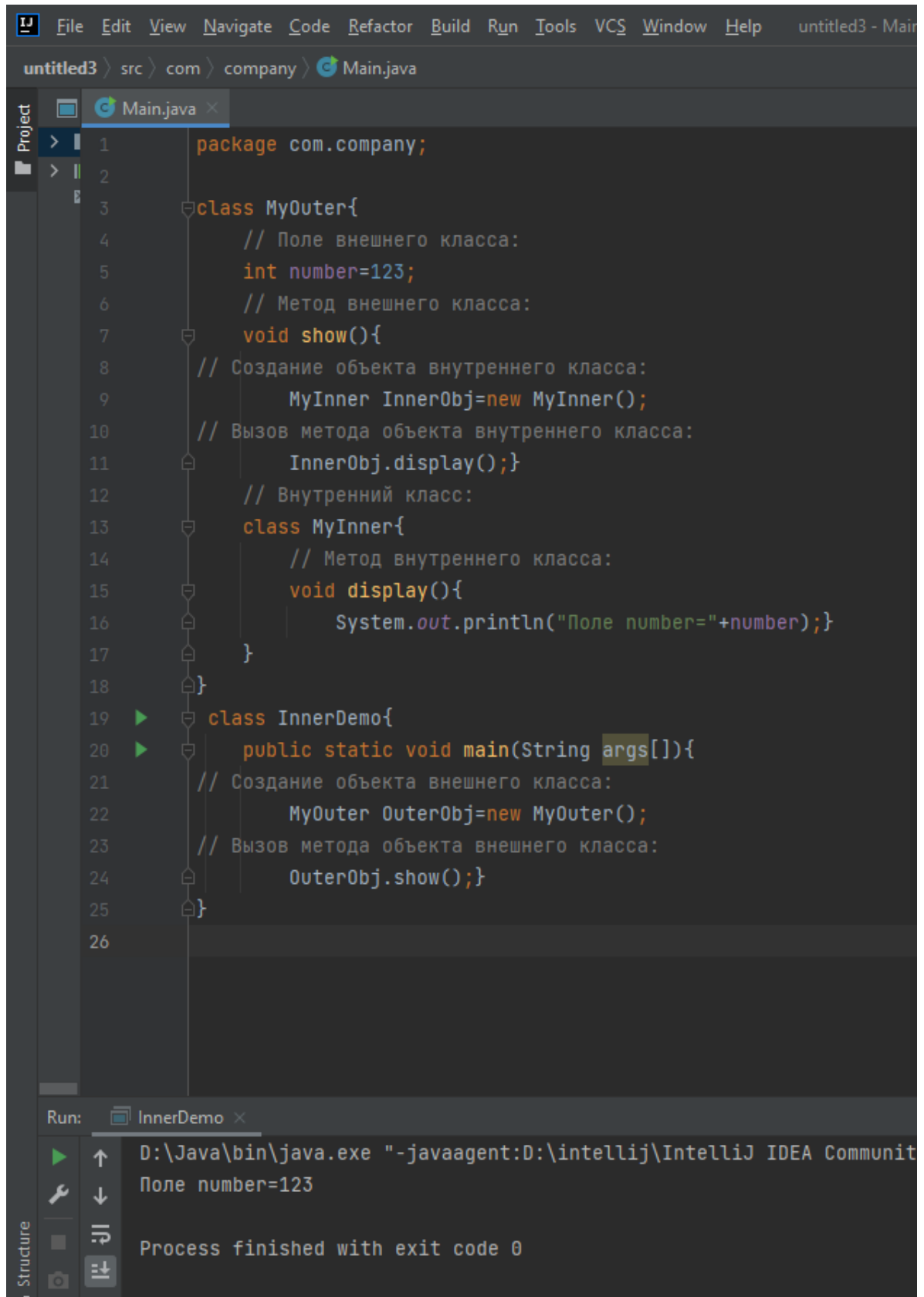


The screenshot displays the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the file path: untitled3 > src > com > company > Main.java. The Project tool window on the left shows the file structure. The main editor displays the following Java code:

```
1 package com.company;
2
3 class MyClass{
4     // Поля класса:
5     double Re,Im;
6     void set(double Re,double Im){
7         // Использование ссылки this:
8         this.Re=Re;
9         this.Im=Im;}
10    void get(){
11        // Инструкция перехода на новую строку \n:
12        System.out.println("Значения полей:\nRe="+this.Re+" и Im="+this.Im);}
13    }
14    class ThisDemo {
15    public static void main(String[] args) {
16        MyClass obj = new MyClass();
17        obj.set(1, 5);
18        obj.get();
19    }
20 }
```

Below the code editor, the Run tool window is open, showing the execution of the 'ThisDemo' class. The command executed is: `D:\Java\bin\java.exe "-javaagent:D:\intelliJ\IntelliJ IDEA Community Edition 2021.2.3\l...`. The output is: `Значения полей:
Re=1.0 и Im=5.0`. The process finished with exit code 0.

Листинг 4.6



Листинг 4.9

```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
untitled3 > src > com > company > Main.java

Project Main.java x
1 package com.company;
2
3 // Класс с математическими функциями:
4 class MyMath{
5     // Интервал разложения в ряд Фурье:
6     static double L=Math.PI;
7     // Экспонента:
8     static double Exp(double x,int N){
9         int i;
10        double s=0,q=1;
11        for(i=0;i<N;i++){
12            s+=q;
13            q*=x/(i+1);}
14        return s+q;}
15    // Синус:
16    static double Sin(double x,int N){
17        int i;
18        double s=0,q=x;
19        for(i=0;i<N;i++){
20            s+=q;
21            q*=(-1)*x*x/(2*i+2)/(2*i+3);}
22        return s+q;}
23    // Косинус:
24    static double Cos(double x,int N){
25        int i;
26        double s=0,q=1;
27        for(i=0;i<N;i++){
28            s+=q;
29            q*=(-1)*x*x/(2*i+1)/(2*i+2);}
30        return s+q;}
31    // Функция Бесселя:
32    static double BesselJ(double x,int N){
33        int i;
34        double s=0,q=1;
35        for(i=0;i<N;i++){
36            s+=q;
37            q*=(-1)*x*x/4/(i+1)/(i+1);}
38        return s+q;}
39    // Ряд Фурье по синусам:
40    static double FourSin(double x,double[] a){

```

Run: MathDemo x

```
Math.java
// Ряд Фурье по синусам:
static double FourSin(double x, double[] a){
    int i, N=a.length;
    double s=0;
    for(i=0; i<N; i++){
        s+=a[i]*Math.sin(Math.PI*x*(i+1)/L);}
    return s;}

// Ряд Фурье по косинусам:
static double FourCos(double x, double[] a){
    int i, N=a.length;
    double s=0;
    for(i=0; i<N; i++){
        s+=a[i]*Math.cos(Math.PI*x*i/L);}
    return s;}
}

class MathDemo{
    public static void main(String args[]){
        System.out.println("Примеры вызова функций:");
        // Вычисление экспоненты:
        System.out.println("exp(1)=" + MyMath.Exp(x: 1, N: 30));
        // Вычисление синуса:
        System.out.println("sin(pi)=" + MyMath.Sin(Math.PI, N: 100));
        // Вычисление косинуса:
        System.out.println("cos(pi/2)=" + MyMath.Cos(x: Math.PI/2, N: 100));
        // Вычисление функции Бесселя:
        System.out.println("J0(mu1)=" + MyMath.BesselJ(x: 2.404825558, N: 100));
        // Заполнение массивов коэффициентов рядов Фурье для функции y(x)=x:
        int m=1000;
        double[] a=new double[m];
        double[] b=new double[m+1];
        b[0]=MyMath.L/2;
        for(int i=1; i<=m; i++){
            a[i-1]=(2*(i%2)-1)*2*MyMath.L/Math.PI/i;
            b[i]=-4*(i%2)*MyMath.L/Math.pow(Math.PI*i, 2);}
        // Вычисление функции y(x)=x через синус-ряд Фурье:
        System.out.println("2.0->" + MyMath.FourSin(x: 2.0, a));
        // Вычисление функции y(x)=x через косинус-ряд Фурье:
        System.out.println("2.0->" + MyMath.FourCos(x: 2.0, b));
    }
}
```

```
MathDemo
D:\Java\bin\java.exe "-javaagent:D:\int
Примеры вызова функций:
exp(1)=2.7182818284590455
sin(pi)=2.4790606536130346E-16
cos(pi/2)=4.590388303752165E-17
J0(mu1)=-1.5793881580131606E-10
2.0->1.9996438367829905
2.0->1.999999349096476

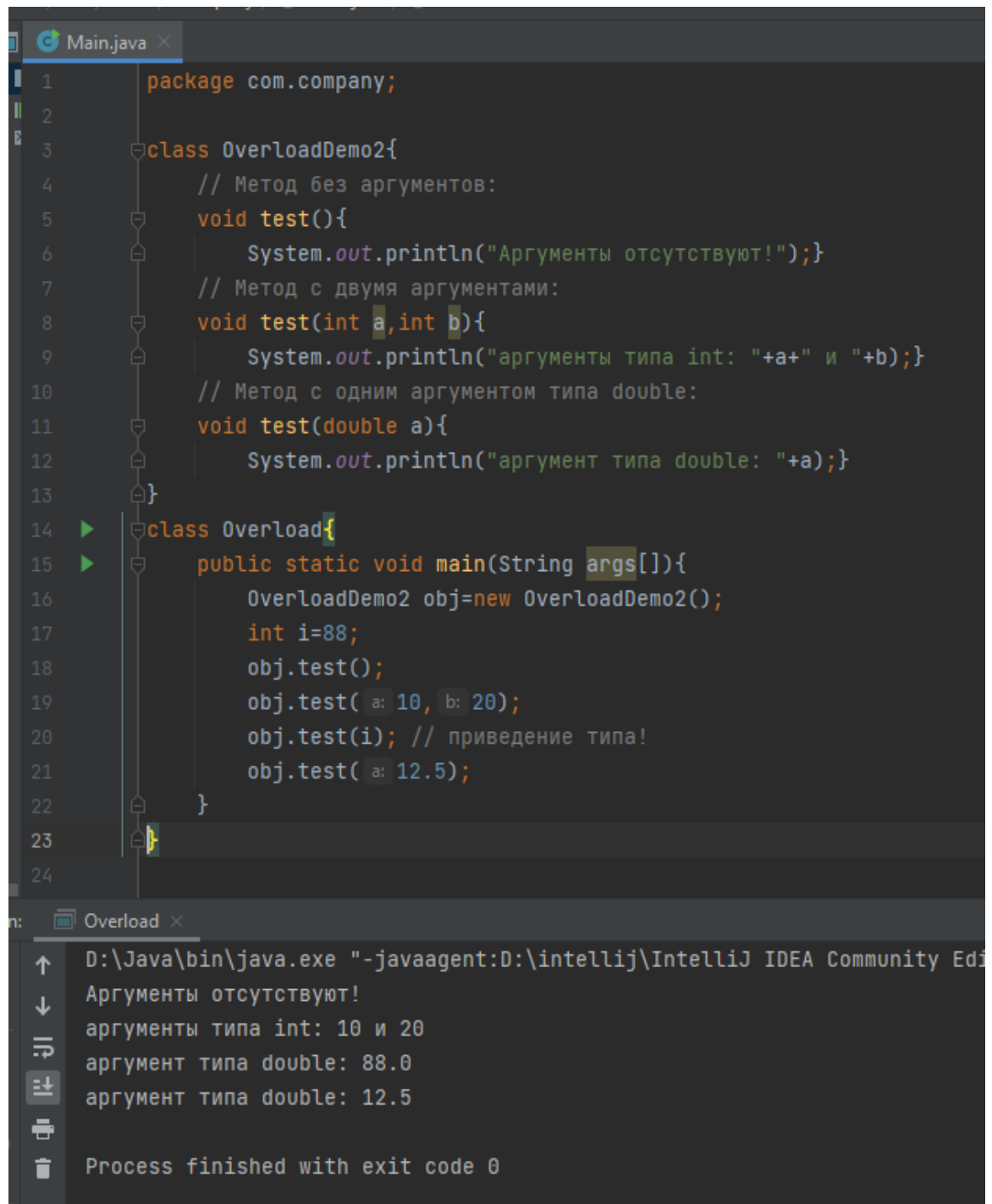
Process finished with exit code 0
```

Листинг 5.1

```
1 package com.company;
2
3 // Класс с перегруженным методом:
4 class OverloadDemo{
5     // Вариант метода без аргументов:
6     void test(){
7         System.out.println("Аргументы отсутствуют!");
8     }
9     // Вариант метода с одним целым аргументом:
10    void test(int a){
11        System.out.println("аргумент типа int: "+a);
12    }
13    // Вариант метода с аргументом типа double и результатом типа double:
14    double test(double a){
15        System.out.println("аргумент типа double: "+a);
16        return a;
17    }
18    // Класс с методом main():
19    class Overload{
20        public static void main(String args[]){
21            OverloadDemo obj=new OverloadDemo();
22            double result;
23            // Вызов перегруженного метода:
24            obj.test();
25            obj.test( a: 10);
26            result=obj.test( a: 12.5);
27            System.out.println("Результат: "+result);
28        }
29    }
30 }
```

↑ D:\Java\bin\java.exe "-javaagent:D:\intelliJ\IntelliJ IDEA Community Edition 2021.
↓ Аргументы отсутствуют!
↕ аргумент типа int: 10
↕ аргумент типа double: 12.5
↕ Результат: 12.5
🖨
🗑 Process finished with exit code 0

Листинг 5.2



```
1 package com.company;
2
3 class OverloadDemo2{
4     // Метод без аргументов:
5     void test(){
6         System.out.println("Аргументы отсутствуют!");}
7     // Метод с двумя аргументами:
8     void test(int a,int b){
9         System.out.println("аргументы типа int: "+a+" и "+b);}
10    // Метод с одним аргументом типа double:
11    void test(double a){
12        System.out.println("аргумент типа double: "+a);}
13    }
14    class Overload{
15    public static void main(String args[]){
16        OverloadDemo2 obj=new OverloadDemo2();
17        int i=88;
18        obj.test();
19        obj.test(a: 10, b: 20);
20        obj.test(i); // приведение типа!
21        obj.test(a: 12.5);
22    }
23
24
```

Output:

```
D:\Java\bin\java.exe "-javaagent:D:\intellij\IntelliJ IDEA Community Edi
Аргументы отсутствуют!
аргументы типа int: 10 и 20
аргумент типа double: 88.0
аргумент типа double: 12.5
Process finished with exit code 0
```