

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <module type="JAVA_MODULE" version="4">
3   <component name="NewModuleRootManager" inherit-compiler-output="true">
4     <exclude-output />
5     <content url="file://$MODULE_DIR$">
6       <sourceFolder url="file://$MODULE_DIR$/src" isTestSource="false" />
7     </content>
8     <orderEntry type="inheritedJdk" />
9     <orderEntry type="sourceFolder" forTests="false" />
10  </component>
11 </module>
```

```
1
2  // IntelliJ API Decompiler stub source generated from a class file
3  // Implementation of methods is not available
4
5  public class Main {
6      public Main() { /* compiled code */ }
7
8      public static void main(java.lang.String[] args) { /* compiled code */ }
9  }
```

```
1
2 // IntelliJ API Decompiler stub source generated from a class file
3 // Implementation of methods is not available
4
5 public abstract class Operate <T extends java.lang.Number> extends Calculator<T> {
6     public Operate() { /* compiled code */ }
7
8     public double add(T first, T second) { /* compiled code */ }
9
10    public abstract double add(double v, double v1);
11
12    public double subtract(T first, T second) { /* compiled code */ }
13
14    public abstract double subtract(double v, double v1);
15
16    public double multiply(T first, T second) { /* compiled code */ }
17
18    public abstract double multiply(double v, double v1);
19
20    public double divide(T first, T second) { /* compiled code */ }
21
22    public abstract double divide(double v, double v1);
23
24    public void calculate(char operation) { /* compiled code */ }
25 }
```

```
1
2 // IntelliJ API Decompiler stub source generated from a class file
3 // Implementation of methods is not available
4
5 public class Calculator <T extends java.lang.Number> {
6     protected T first;
7     protected T second;
8
9     public Calculator() { /* compiled code */ }
10
11    public T getFirst() { /* compiled code */ }
12
13    public void setFirst(T first) { /* compiled code */ }
14
15    public T getSecond() { /* compiled code */ }
16
17    public void setSecond(T second) { /* compiled code */ }
18 }
```

```

1 public class Main {
2     public static void main (String[] args) {
3         Calculator <Double> operation = new Operate <Double> () {
4             @Override
5             public double add(double first, double second) {
6                 return first + second;
7             }
8             public double subtract(double first, double second) {
9                 return first - second;
10            }
11            public double multiply(double first, double second) {
12                return first * second;
13            }
14            public double divide(double first, double second) {
15                return first / second;
16            }
17        };
18        // Ispis anonimne klase
19        System.out.println(((Operate<Double>)operation).add(12.0,2.0));
20        System.out.println(((Operate<Double>)operation).subtract(12.0,2.0));
21        System.out.println(((Operate<Double>)operation).multiply(12.0,2.0));
22        System.out.println(((Operate<Double>)operation).divide(12.0,2.0));
23
24        //      Down-casting
25        //      Operate operator = (Operate) operation;
26        //      Scanner sc = new Scanner(System.in);
27        //      boolean calculator = true;
28        //
29        //      while (calculator) {
30        //          System.out.println("Unesite prvi broj: ");
31        //          operation.setFirst(sc.nextDouble());
32        //
33        //          System.out.println("Unesite drugi broj: ");
34        //          operation.setSecond(sc.nextDouble());
35        //
36        //          System.out.println("Unesite zeljeni operator: ");
37        //          operator.calculate(sc.next().charAt(0));
38        //
39        //          System.out.println("\nZelite li jos koju operaciju izvest? (da/ne)");
40        //          String decider = sc.next();
41        //          if (decider.equalsIgnoreCase("da")) calculator = true;
42        //          else calculator = false;
43        //      }
44    }
45 }

```

```
1 public abstract class Operate <T extends Number> extends Calculator <T> {
2     public double add (T first, T second) {
3         // Unboxing
4         return first.doubleValue() + second.doubleValue();
5     }
6     public abstract double add(double first, double second);
7
8     public double subtract (T first, T second) {
9         // Unboxing
10        return first.doubleValue() - second.doubleValue();
11    }
12    public abstract double subtract(double first, double second);
13
14    public double multiply (T first, T second) {
15        // Unboxing
16        return first.doubleValue() * second.doubleValue();
17    }
18    public abstract double multiply(double first, double second);
19
20    public double divide (T first, T second) {
21        // Unboxing
22        return first.doubleValue() / second.doubleValue();
23    }
24    public abstract double divide(double first, double second);
25
26
27    public void calculate (char operation) {
28        System.out.println("\n" + first + " " + operation + " " + second + ": ");
29
30        switch (operation) {
31            case ('+'):
32                System.out.println(add(first, second));
33                break;
34            case ('-'):
35                System.out.println(subtract(first, second));
36                break;
37            case ('*'):
38                System.out.println(multiply(first, second));
39                break;
40            case ('/'):
41                System.out.println(divide(first, second));
42                break;
43            default:
44                System.out.println("Krivo unesen operator.");
45        }
46    }
47
48
49
50 }
```

```
1 public class Calculator <T extends Number> {
2     protected T first;
3     protected T second;
4
5     public T getFirst() {
6         return first;
7     }
8
9     public void setFirst(T first) {
10        this.first = first;
11    }
12
13    public T getSecond() {
14        return second;
15    }
16
17    public void setSecond(T second) {
18        this.second = second;
19    }
20 }
21
```

```
1 "C:\Program Files\BellSoft\LibericaJDK-19-Full\bin\java.exe" "-javaagent:C:\Program Files\
JetBrains\IntelliJ IDEA 2022.2.3\lib\idea_rt.jar=49581:C:\Program Files\JetBrains\IntelliJ
IDEA 2022.2.3\bin" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=
UTF-8 -classpath D:\Faks\SuvremeneTehnikeProgramiranja\Zadace\Zadaca07b\out\production\
Zadaca07b Main
2 14.0
3 10.0
4 24.0
5 6.0
6
7 Process finished with exit code 0
8
```