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

```java

  // IntelliJ API Decompiler stub source generated from a class file
  // Implementation of methods is not available

public class Main {
    public Main() { /* compiled code */ }

    public static void main(java.lang.String[] args) { /* compiled code */ }
}
```

```java

  // IntelliJ API Decompiler stub source generated from a class file
  // Implementation of methods is not available

public abstract class Operate <T extends java.lang.Number> extends Calculator<T> {
    public Operate() { /* compiled code */ }

    public double add(T first, T second) { /* compiled code */ }

    public abstract double add(double v, double v1);

    public double subtract(T first, T second) { /* compiled code */ }

    public abstract double subtract(double v, double v1);

    public double multiply(T first, T second) { /* compiled code */ }

    public abstract double multiply(double v, double v1);

    public double divide(T first, T second) { /* compiled code */ }

    public abstract double divide(double v, double v1);

    public void calculate(char operation) { /* compiled code */ }
}
```

```java

  // IntelliJ API Decompiler stub source generated from a class file
  // Implementation of methods is not available

public class Calculator <T extends java.lang.Number> {
    protected T first;
    protected T second;

    public Calculator() { /* compiled code */ }

    public T getFirst() { /* compiled code */ }

    public void setFirst(T first) { /* compiled code */ }

    public T getSecond() { /* compiled code */ }

    public void setSecond(T second) { /* compiled code */ }
}
```

```java
public class Main {
    public static void main (String[] args) {
        Calculator <Double> operation = new Operate <Double> () {
            @Override
            public double add(double first, double second) {
                return first + second;
            }
            public double subtract(double first, double second) {
                return first - second;
            }
            public double multiply(double first, double second) {
                return first * second;
            }
            public double divide(double first, double second) {
                return first / second;
            }
        };
        // Ispis anonimne klase
        System.out.println(((Operate<Double>)operation).add(12.0,2.0));
        System.out.println(((Operate<Double>)operation).subtract(12.0,2.0));
        System.out.println(((Operate<Double>)operation).multiply(12.0,2.0));
        System.out.println(((Operate<Double>)operation).divide(12.0,2.0));

//        Down-casting
//        Operate operator = (Operate) operation;
//        Scanner sc = new Scanner(System.in);
//        boolean calculator = true;
//
//        while (calculator) {
//            System.out.println("Unesite prvi broj: ");
//            operation.setFirst(sc.nextDouble());
//
//            System.out.println("Unesite drugi broj: ");
//            operation.setSecond(sc.nextDouble());
//
//            System.out.println("Unesite zeljeni operator: ");
//            operator.calculate(sc.next().charAt(0));
//
//            System.out.println("\nZelite li jos koju operaciju izvest? (da/ne)");
//            String decider = sc.next();
//            if (decider.equalsIgnoreCase("da")) calculator = true;
//            else calculator = false;
//        }
    }
}
```

```java
public abstract class Operate <T extends Number> extends Calculator <T> {
    public double add (T first, T second) {
        // Unboxing
        return first.doubleValue() + second.doubleValue();
    }
    public abstract double add(double first, double second);

    public double subtract (T first, T second) {
        // Unboxing
        return first.doubleValue() - second.doubleValue();
    }
    public abstract double subtract(double first, double second);

    public double multiply (T first, T second) {
        // Unboxing
        return first.doubleValue() * second.doubleValue();
    }
    public abstract double multiply(double first, double second);

    public double divide (T first, T second) {
        // Unboxing
        return first.doubleValue() / second.doubleValue();
    }
    public abstract double divide(double first, double second);


    public void calculate (char operation) {
        System.out.println("\n" + first + " " + operation + " " + second + ": ");

        switch (operation) {
            case ('+'):
                System.out.println(add(first, second));
                break;
            case ('-'):
                System.out.println(subtract(first, second));
                break;
            case ('*'):
                System.out.println(multiply(first, second));
                break;
            case ('/'):
                System.out.println(divide(first, second));
                break;
            default:
                System.out.println("Krivo unesen operator.");
        }
    }



}
```

```java
public class Calculator <T extends  Number> {
    protected T first;
    protected T second;

    public T getFirst() {
        return first;
    }

    public void setFirst(T first) {
        this.first = first;
    }

    public T getSecond() {
        return second;
    }

    public void setSecond(T second) {
        this.second = second;
    }
}

```

```
"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
14.0
10.0
24.0
6.0

Process finished with exit code 0

```