

Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №3
по дисциплине “Современные платформы программирования”

Выполнил:

Студент 3 курса

Группы ПО-8

Бондаренко К.А.

Проверил:

Крощенко А.А.

Цель работы: научиться создавать и использовать классы в программах на языке программирования Java.

Ход работы

Вариант 1

Задание 1. Равнобедренный треугольник, заданный длинами сторон – Предусмотреть возможность определения площади и периметра, а также логический метод, определяющий существует ли такой треугольник. Конструктор должен позволить создавать объекты с начальной инициализацией. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Код программы:

Класс равнобедренного треугольника:

```
public class IsoscelesTriangle {
    private double sideA, sideB, sideC;

    public IsoscelesTriangle(double sideA, double sideB, double sideC) {
        this.sideA = sideA;
        this.sideB = sideB;
        this.sideC = sideC;
    }

    public double getSideA() {
        return sideA;
    }
    public double getSideB() {
        return sideB;
    }
    public double getSideC() {
        return sideC;
    }

    public void setSideA(double sideA) {
        this.sideA = sideA;
    }
    public void setSideB(double sideB) {
        this.sideB = sideB;
    }
    public void setSideC(double sideC) {
        this.sideC = sideC;
    }

    public double determinePerimeter() {
        return sideA + sideB + sideC;
    }
}
```

```

    public double determineArea() {
        double semiperimeter = (sideA + sideB + sideC) / 2;
        return Math.sqrt(semiperimeter * (semiperimeter - sideA) * (semiperimeter - sideB) *
(semiperimeter - sideC));
    }

    public boolean isExists() { return sideA + sideB > sideC && sideA + sideC > sideB && sideB + sideC >
sideA; }

    public boolean isIsoscelesTriangle() {
        return (sideA == sideB && sideB != sideC) || (sideA == sideC && sideC != sideB) || (sideB == sideC
&& sideC != sideA);
    }

    @Override
    public String toString() {
        String text = "sideA: " + sideA + "\nsideB: " + sideB + "\nsideC: " + sideC;
        return super.toString();
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }

        if (obj == null || this.getClass() != obj.getClass()) {
            return false;
        }

        IsoscelesTriangle other = (IsoscelesTriangle) obj;
        return sideA == other.sideA && sideB == other.sideB && sideC == other.sideC;
    }
}

```

Класс с методом main:

```

public class Main {
    public static void main(String[] args) {
        IsoscelesTriangle triangle1 = new IsoscelesTriangle(3, 7, 4);
        IsoscelesTriangle triangle2 = new IsoscelesTriangle(3, 4, 4);

        System.out.println("Треугольник 1: " + triangle1);
        System.out.println("Периметр треугольника 1: " + triangle1.determinePerimeter());
        System.out.println("Площадь треугольника 1: " + triangle1.determineArea());
        System.out.println("Существует ли треугольник 1? " + triangle1.isExists());
        System.out.println("Является ли треугольник 1 равнобедренным? " +
triangle1.isIsoscelesTriangle());

        System.out.println("\nТреугольник 2: " + triangle2);
        System.out.println("Периметр треугольника 2: " + triangle2.determinePerimeter());
        System.out.println("Площадь треугольника 2: " + triangle2.determineArea());
        System.out.println("Существует ли треугольник 2? " + triangle2.isExists());
        System.out.println("Является ли треугольник 2 равнобедренным? " +

```

```

triangle2.isIsoscelesTriangle());

        System.out.println("\nРавны ли треугольник 1 и треугольник 2? " + triangle1.equals(triangle2));
    }
}

```

Результаты работы программы:

```

Треугольник 1: IsoscelesTriangle@133314b
Периметр треугольника 1: 14.0
Площадь треугольника 1: 0.0
Существует ли треугольник 1? false
Является ли треугольник 1 равнобедренным? false

Треугольник 2: IsoscelesTriangle@79fc0f2f
Периметр треугольника 2: 11.0
Площадь треугольника 2: 5.562148865321747
Существует ли треугольник 2? true
Является ли треугольник 2 равнобедренным? true

Равны ли треугольник 1 and треугольник 2? false

```

Задание 2. Написать стековый калькулятор, который принимает в качестве аргумента командой строки имя файла, содержащего команды. Если аргумента нет, то использовать стандартный поток ввода для чтения команд. Для вычислений допускается использовать вещественные числа.

Реализовать следующий набор команд:

- # – строка с комментарием.
- POP , PUSH – снять/положить число со/на стек(а).
- + , - , * , / , SQRT – арифметические операции. Используют один или два верхних элемента стека, изымают их из стека, помещая результат назад
- PRINT – печать верхнего элемента стека (без удаления).
- DEFINE – задать значение параметра. В дальнейшем везде использовать вместо параметра это значение.

Содержимое стека и список определенных именованных параметров передавать команде в виде специального объекта – контекста исполнения. Разработать группу классов исключений, которые будут выбрасывать команды при исполнении. В случае возникновения исключения – выводить информацию об ошибке и продолжать исполнение программы (из файла или команд, вводимых с консоли)


```

case "+":
    if (stack.size() < 2) {
        throw new OperationExecutionException("Недостаточно операндов в стеке!");
    }
    Double a = stack.pop();
    Double b = stack.pop();
    stack.push(a + b);
    break;
case "-":
    if (stack.size() < 2) {
        throw new OperationExecutionException("Недостаточно операндов в стеке!");
    }
    a = stack.pop();
    b = stack.pop();
    stack.push(a - b);
    break;
case "*":
    if (stack.size() < 2) {
        throw new OperationExecutionException("Недостаточно операндов в стеке!");
    }
    a = stack.pop();
    b = stack.pop();
    stack.push(a * b);
    break;
case "/":
    if (stack.size() < 2) {
        throw new OperationExecutionException("Недостаточно операндов в стеке!");
    }
    a = stack.pop();
    b = stack.pop();

    if (b == 0) {
        throw new OperationExecutionException("Деление на 0!");
    }
    stack.push(a / b);
    break;
case "SQRT":
    if (stack.isEmpty()) {
        throw new OperationExecutionException("Стек пуст!");
    }
    a = stack.pop();

    if (a < 0) {
        throw new OperationExecutionException("Извлечение корня из отрицательного
числа!");
    }
    stack.push(Math.sqrt(a));
    break;
case "PRINT":
    if (stack.isEmpty()) {
        throw new OperationExecutionException("Стек пуст!");
    }
    System.out.println(stack.peek());
    break;

```

```

        case "DEFINE":
            defineParameters.put(params[1], Double.parseDouble(params[2]));
            break;
        default:
            throw new CommandParseException("Неизвестная операция:" + operation + "!");
    }
} catch (CalculatorException e) {
    System.out.println("Ошибка:" + e);
}
}
}

```

Класс с методом main:

```

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

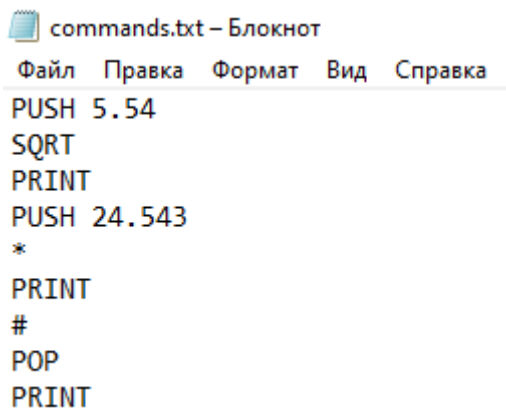
public class Task2 {
    public static void main(String[] args) {
        Calculator calculator = new Calculator();

        try (BufferedReader reader = args.length > 0 ? new BufferedReader(new FileReader(args[0])) :
            new BufferedReader(new InputStreamReader(System.in))) {
            String line;
            while ((line = reader.readLine()) != null) {
                calculator.performCommand(line);
            }
        } catch (IOException e) {
            System.out.println("Ошибка чтения файла: " + e);
        }
    }
}

```

Результаты работы программы:

Содержимое файла:



```

commands.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
PUSH 5.54
SQRT
PRINT
PUSH 24.543
*
PRINT
#
POP
PRINT

```

Тест программы:

```
PS C:\Users\bka32\GitHub-repos\SPP\SPPLab3\Task2\src> javac Task2.java
PS C:\Users\bka32\GitHub-repos\SPP\SPPLab3\Task2\src> java Task2 commands.txt
2.353720459187964
57.7673612298502
Ошибка:OperationExecutionException: Стек пуст!
```

Вывод: научились создавать и использовать классы в программах на языке программирования Java.