

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

Лабораторная работа 3
По дисциплине “СПП”

Выполнил: Вальчук А.А.
Проверил: Крощенко А.А.

Брест 2021

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

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

```
public class IsoscelesTriangle {
    private double legLength;
    private double baseLength;

    public IsoscelesTriangle(double legLength, double baseLength) throws InvalidTriangleException
    {
        ThrowExceptionIfSideLengthIsInvalid(legLength);
        ThrowExceptionIfSideLengthIsInvalid(baseLength);
        ThrowExceptionIfTriangleIsInvalid(legLength, baseLength);

        this.legLength = legLength;
        this.baseLength = baseLength;
    }

    public double getLegLength() {
        return legLength;
    }

    public double getBaseLength() {
        return baseLength;
    }

    public void setLegLength(double legLength) throws InvalidTriangleException {
        ThrowExceptionIfSideLengthIsInvalid(legLength);
        ThrowExceptionIfTriangleIsInvalid(legLength, baseLength);

        this.legLength = legLength;
    }

    public void setBaseLength(double baseLength) throws InvalidTriangleException {
        ThrowExceptionIfSideLengthIsInvalid(baseLength);
        ThrowExceptionIfTriangleIsInvalid(legLength, baseLength);

        this.baseLength = baseLength;
    }
}
```

```

public double getPerimeter() {
    return legLength * 2 + baseLength;
}

public double getArea() {
    return (baseLength / 2) * Math.sqrt(legLength * legLength - baseLength * baseLength / 4);
}

@Override
public String toString() {
    return String.format("IsoscelesTriangle{legLength=%f, baseLength=%f}", legLength,
baseLength);
}

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

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

    IsoscelesTriangle isoscelesTriangle = (IsoscelesTriangle) obj;

    return Double.compare(isoscelesTriangle.legLength, legLength) == 0 &&
        Double.compare(isoscelesTriangle.baseLength, baseLength) == 0;
}

private boolean isTriangleValid(double legLength, double baseLength) {
    return legLength * 2 > baseLength;
}

private void ThrowExceptionIfSideLengthIsInvalid(double sideLength) {
    if (sideLength <= 0) {
        throw new IllegalArgumentException("Side length must be greater than 0. Provided value: "
+ sideLength);
    }
}

private void ThrowExceptionIfTriangleIsInvalid(double legLength, double baseLength) throws
InvalidTriangleException {
    if (!isTriangleValid(legLength, baseLength)) {
        throw new InvalidTriangleException("Isosceles triangle is valid if sum of its legs is greater
than its base. " +

```

```

        String.format("Provided values: leg length: %f, base length: %f", legLength,
baseLength));
    }
}

```

```

public class InvalidTriangleException extends Exception {
    public InvalidTriangleException(String message) {
        super(message);
    }
}

```

```

public class Main {

    public static void main(String[] args) throws InvalidTriangleException {
        IsoscelesTriangle firstTriangle = new IsoscelesTriangle(13, 24);

        System.out.println(firstTriangle.getLegLength());
        System.out.println(firstTriangle.getBaseLength());
        System.out.println(firstTriangle.getPerimeter());
        System.out.println(firstTriangle.getArea());
        System.out.println(firstTriangle);

        IsoscelesTriangle secondTriangle = new IsoscelesTriangle(13, 24);
        IsoscelesTriangle thirdTriangle = new IsoscelesTriangle(14, 24);

        System.out.println(firstTriangle.equals(secondTriangle));
        System.out.println(firstTriangle.equals(thirdTriangle));
    }
}

```

```

24.0
50.0
60.0
IsoscelesTriangle{legLength=13,000000, baseLength=24,000000}
true
false

```

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

```
public class StackCalculator {
    private final Stack<Character> operatorStack = new Stack<>();
    private final Stack<Double> valueStack = new Stack<>();

    private boolean isOperator(char ch) {
        return ch == '+' || ch == '-' || ch == '*' || ch == '/';
    }

    private int getPrecedence(char operator) {
        return switch (operator) {
            case '+', '-' -> 1;
            case '*', '/' -> 2;
            default -> throw new IllegalStateException("Unexpected value: " + operator);
        };
    }

    private void processOperator(char operator) throws InvalidExpressionException {
        if (valueStack.size() < 2) {
            throw new InvalidExpressionException();
        }

        double b = valueStack.pop();
        double a = valueStack.pop();

        double result = switch (operator) {
            case '+' -> a + b;
            case '-' -> a - b;
            case '*' -> a * b;
            case '/' -> a / b;
            default -> throw new IllegalStateException("Unexpected value: " + operator);
        };

        valueStack.push(result);
    }

    public double processExpression(String expression) throws InvalidExpressionException {
        for (int i = 0; i < expression.length(); i++) {
            if (expression.charAt(i) >= '0' && expression.charAt(i) <= '9') {
                int valueFirstPosition = i;
```

```

        while (i + 1 < expression.length() && (expression.charAt(i + 1) >= '0' &&
expression.charAt(i + 1) <= '9' || expression.charAt(i + 1) == '.')) {
            i++;
        }

        double value = Double.parseDouble(expression.substring(valueFirstPosition, i + 1));
        valueStack.push(value);
    }
    else if (isOperator(expression.charAt(i))) {
        while (!operatorStack.empty() && isOperator(operatorStack.peek()) &&
getPrecedence(expression.charAt(i)) <= getPrecedence(operatorStack.peek())) {
            processOperator(operatorStack.pop());
        }
        operatorStack.push(expression.charAt(i));
    }
    else if (expression.charAt(i) == '(') {
        operatorStack.push(expression.charAt(i));
    }
    else if (expression.charAt(i) == ')') {
        while (!operatorStack.empty() && isOperator(operatorStack.peek())) {
            processOperator(operatorStack.pop());
        }
        if (!operatorStack.empty() && operatorStack.peek() == '(') {
            operatorStack.pop();
        }
        else {
            throw new InvalidExpressionException("Error: unbalanced parenthesis");
        }
    }
    else if (expression.charAt(i) != ' ') {
        throw new InvalidExpressionException();
    }
}

while (!operatorStack.empty() && isOperator(operatorStack.peek())) {
    processOperator(operatorStack.pop());
}

double result = valueStack.peek();
valueStack.pop();

if (!operatorStack.empty() || !valueStack.empty()) {
    throw new InvalidExpressionException();
}

```

```

        return result;
    }
}

```

```

public class InvalidExpressionException extends Exception {
    public InvalidExpressionException() {
        super("Math expression exception.");
    }
    public InvalidExpressionException(String message) {
        super(message);
    }
}

```

```

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.Scanner;

```

```

public final class Utilities {
    public static String getMathExpression(String fileName) throws IOException {
        if (fileName == null) {
            Scanner scanner = new Scanner(System.in);

            return scanner.nextLine();
        }

        return Files.readString(Path.of(fileName));
    }
}

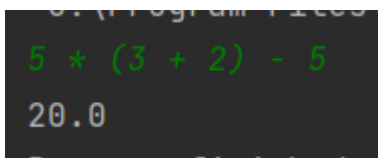
```

```

public class Main {

    public static void main(String[] args) throws IOException, InvalidExpressionException {
        String mathExpression = Utilities.getMathExpression(args.length == 0 ? null : args[0]);
        StackCalculator stackCalculator = new StackCalculator();
        System.out.print(stackCalculator.processExpression(mathExpression));
    }
}

```



```

5 * (3 + 2) - 5
20.0

```

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