

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
Шлыков А.Л.
Проверил:
Крощенко А.А.

Брест 2024

Цель работы:

научиться создавать и использовать классы в программах на языке программирования Java

Задание 1

Реализовать простой класс.

Требования к выполнению

- Реализовать пользовательский класс по варианту.
- Создать другой класс с методом main, в котором будут находиться примеры использования пользовательского класса.

Для каждого класса

- Создать поля классов
- Создать методы классов
- Добавьте необходимые get и set методы (по необходимости)
- Укажите соответствующие модификаторы видимости
- Добавьте конструкторы
- Переопределить методы toString() и equals()

1)Равнобедренный треугольник, заданный длинами сторон – Предусмотреть возможность определения площади и периметра, а так же логический метод, определяющий существует ли такой треугольник. Конструктор должен позволить создавать объекты с начальной

инициализацией. Реализовать метод equals, выполняющий сравнение объектов данного типа.

Класс равнобедренный треугольника:

```
public class Triangle {  
    private double sideA;  
    private double sideB;  
    private double sideC;  
  
    public Triangle(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 double getPerimeter() {
    return sideA + sideB + sideC;
}

public double getArea() {
    double s = getPerimeter() / 2;
    return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
}

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

@Override
public boolean equals(Object obj) {
    if (this == obj) return true;
    if (obj == null || getClass() != obj.getClass()) return false;
    Triangle triangle = (Triangle) obj;
    return Double.compare(triangle.sideA, sideA) == 0 &&
        Double.compare(triangle.sideB, sideB) == 0 &&
        Double.compare(triangle.sideC, sideC) == 0;
}

@Override
public String toString() {
    return "Triangle{" +
        "sideA=" + sideA +
        ", sideB=" + sideB +
        ", sideC=" + sideC +
        '}';
}
}

```

Main:

```

public class laba3_1 {
    public static void main(String[] args) {
        Triangle triangle = new Triangle(3, 4, 5);
        System.out.println("Triangle: " + triangle);
        System.out.println("Perimeter: " + triangle.getPerimeter());
        System.out.println("Area: " + triangle.getArea());
        System.out.println("Is valid: " + triangle.isValid());
    }
}

```

```

[Running] cd "/Users/harweast/Documents/3kurs/sem2/SPP/laba3/3/src/" && javac laba3_1.java && java laba3_1
Triangle: Triangle{sideA=3.0, sideB=4.0, sideC=5.0}
Perimeter: 12.0
Area: 6.0
Is valid: true

[Done] exited with code=0 in 0.353 seconds

```

Задание 2

Разработать автоматизированную систему на основе некоторой структуры данных, манипулирующей объектами пользовательского класса. Реализовать требуемые функции обработки данных

Требования к выполнению

- Задание посвящено написанию классов, решающих определенную задачу автоматизации;
- Данные для программы загружаются из файла (формат произволен). Файл создать и написать вручную.

1) Стековый калькулятор

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)

Калькулятор:

```
public class StackCalculator {
    private final Deque<Double> stack = new ArrayDeque<>();
    private final Map<String, Double> variables = new HashMap<>();

    public void execute(String command) {
        String[] parts = command.split(" ");
        switch (parts[0]) {
            case "PUSH":
                stack.push(Double.parseDouble(parts[1]));
                break;
            case "POP":
                if (stack.isEmpty()) {
                    throw new RuntimeException("Stack is empty");
                }
                stack.pop();
                break;
            case "+":
                checkStackSize(2);
                stack.push(stack.pop() + stack.pop());
                break;
            case "-":
                checkStackSize(2);
                stack.push(-stack.pop() + stack.pop());
                break;
            case "*":
                checkStackSize(2);
                stack.push(stack.pop() * stack.pop());
                break;
            case "/":
                checkStackSize(2);
                double divisor = stack.pop();
                if (divisor == 0) {
                    throw new ArithmeticException("Division by zero");
                }
                stack.push(stack.pop() / divisor);
                break;
            case "SQRT":
                checkStackSize(1);
                stack.push(Math.sqrt(stack.pop()));
                break;
            case "PRINT":
                if (stack.isEmpty()) {
                    throw new RuntimeException("Stack is empty");
                }
                System.out.println(stack.peek());
                break;
            case "DEFINE":
                variables.put(parts[1], Double.parseDouble(parts[2]));
                break;
            default:
                throw new IllegalArgumentException("Unknown command: " + parts[0]);
        }
    }

    private void checkStackSize(int size) {
        if (stack.size() < size) {
            throw new RuntimeException("Stack contains less than " + size + " elements");
        }
    }
}
```

Main:

```
public class laba3_2 {  
    public static void main(String[] args) {  
        StackCalculator calculator = new StackCalculator();  
        Scanner scanner = new Scanner(System.in);  
        while (scanner.hasNextLine()) {  
            try {  
                calculator.execute(scanner.nextLine());  
            } catch (Exception e) {  
                System.out.println("Error: " + e.getMessage());  
            }  
        }  
        scanner.close();  
    }  
}
```

```
harweast@MacBook-Air-2 src % java laba3_2  
PUSH 5  
PUSH 6  
+  
PRINT  
11.0
```

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