МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**Кафедра ''САПР''**

Отчет о выполненной лабораторной работе № 1

по дисциплине: “Объектно-ориентированное программирование”

Выполнили ст.гр. 20ВА1:

Комаров М.В.

Пачков А.В.

Принял:

Подмарькова Е.М.

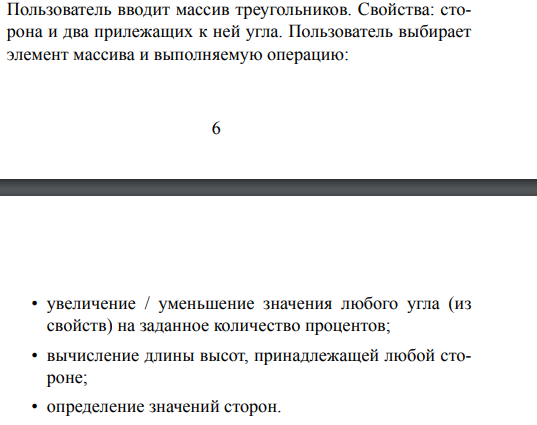
Пенза, 2024

**Цель работы:**

Ознакомиться с процессом классификации на примере моделирования классов предметной области.

**Ход работы:**

Задание:



Код программы:

import java.util.Scanner;

class Triangle {

private double sideA; // Сторона A треугольника

private double angleAB; // Угол AB в градусах

private double angleBC; // Угол BC в градусах

// Конструктор для создания треугольника с заданной стороной и углами

public Triangle(double sideA, double angleAB, double angleBC) {

this.sideA = sideA;

this.angleAB = angleAB;

this.angleBC = angleBC;

}

// Метод для проверки корректности треугольника

public boolean isValid() {

double angleCA = 180 - (angleAB + angleBC); // Вычисление третьего угла

// Проверка, что все углы положительные и сумма углов равна 180 градусам

if (angleAB <= 0 || angleBC <= 0 || angleCA <= 0) {

return false;

}

return true;

}

// Метод для увеличения угла AB на заданный процент

public void increaseAngleAB(double percent) {

angleAB += angleAB \* percent / 100;

// Ограничение, чтобы угол не стал некорректным

if (angleAB >= 180) {

angleAB = 179.99;

}

}

// Метод для уменьшения угла AB на заданный процент

public void decreaseAngleAB(double percent) {

angleAB -= angleAB \* percent / 100;

// Ограничение, чтобы угол не стал некорректным

if (angleAB <= 0) {

angleAB = 0.01;

}

}

// Метод для вычисления высоты, опущенной к стороне A

public double calculateHeightToSideA() {

double angleBCInRadians = Math.toRadians(angleBC); // Преобразование угла в радианы

return sideA \* Math.sin(angleBCInRadians); // Вычисление высоты

}

// Геттер для стороны A

public double getSideA() {

return sideA;

}

// Геттер для угла AB

public double getAngleAB() {

return angleAB;

}

// Геттер для угла BC

public double getAngleBC() {

return angleBC;

}

// Метод для вычисления остальных сторон треугольника с использованием закона синусов

public double[] calculateOtherSides() {

double angleCA = 180 - (angleAB + angleBC); // Вычисление третьего угла

double angleCAInRadians = Math.toRadians(angleCA);

double angleABInRadians = Math.toRadians(angleAB);

double angleBCInRadians = Math.toRadians(angleBC);

// Вычисление сторон B и C

double sideB = (sideA \* Math.sin(angleBCInRadians)) / Math.sin(angleCAInRadians);

double sideC = (sideA \* Math.sin(angleABInRadians)) / Math.sin(angleCAInRadians);

return new double[]{sideB, sideC};

}

// Переопределение метода toString() для вывода информации о треугольнике

@Override

public String toString() {

double[] sides = calculateOtherSides(); // Вычисление остальных сторон

return "Triangle{" +

"sideA=" + sideA +

", sideB=" + sides[0] +

", sideC=" + sides[1] +

", angleAB=" + angleAB +

", angleBC=" + angleBC +

'}';

}

}

public class TriangleOperations {

public static void main(String[] args) {

Scanner = new Scanner(System.in);

// Запрос количества треугольников у пользователя

System.out.println("Введите количество треугольников: ");

int numTriangles = scanner.nextInt();

Triangle[] triangles = new Triangle[numTriangles]; // Создание массива треугольников

// Ввод параметров для каждого треугольника

for (int i = 0; i < numTriangles; i++) {

System.out.println("Введите параметры треугольника " + (i + 1) + ": ");

System.out.println("Сторона A: ");

double sideA = scanner.nextDouble();

System.out.println("Угол AB: ");

double angleAB = scanner.nextDouble();

System.out.println("Угол BC: ");

double angleBC = scanner.nextDouble();

triangles[i] = new Triangle(sideA, angleAB, angleBC); // Создание треугольника

// Проверка корректности треугольника

if (!triangles[i].isValid()) {

System.out.println("Некорректные параметры для треугольника. Попробуйте снова.");

i--; // Повтор ввода в случае некорректных данных

}

}

// Выбор треугольника для выполнения операций

System.out.println("Выберите треугольник (от 0 до " + (numTriangles - 1) + "): ");

int index = scanner.nextInt();

if (index < 0 || index >= triangles.length) {

System.out.println("Некорректный индекс треугольника.");

return;

}

Triangle selectedTriangle = triangles[index]; // Выбранный треугольник

System.out.println("Выбранный треугольник: " + selectedTriangle);

// Выбор операции для выполнения

System.out.println("Выберите операцию: ");

System.out.println("1 - Увеличить угол AB");

System.out.println("2 - Уменьшить угол AB");

System.out.println("3 - Вычислить высоту к стороне A");

System.out.println("4 - Вывести все стороны треугольника");

int operation = scanner.nextInt();

// Выполнение выбранной операции

switch (operation) {

case 1:

System.out.println("Введите процент увеличения: ");

double increasePercent = scanner.nextDouble();

selectedTriangle.increaseAngleAB(increasePercent);

break;

case 2:

System.out.println("Введите процент уменьшения: ");

double decreasePercent = scanner.nextDouble();

selectedTriangle.decreaseAngleAB(decreasePercent);

break;

case 3:

double height = selectedTriangle.calculateHeightToSideA();

System.out.println("Высота к стороне A: " + height);

break;

case 4:

double[] sides = selectedTriangle.calculateOtherSides();

System.out.println("Сторона A: " + selectedTriangle.getSideA());

System.out.println("Сторона B: " + sides[0]);

System.out.println("Сторона C: " + sides[1]);

break;

default:

System.out.println("Некорректная операция.");

return;

}

// Проверка корректности треугольника после изменений

if (selectedTriangle.isValid()) {

System.out.println("Измененный треугольник: " + selectedTriangle);

} else {

System.out.println("После изменений треугольник стал некорректным.");

}

}

}

Результат:

