МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по практической работе № 1**

**по учебной дисциплине «Системное программирование»**

**Тема: «Проецирование классов»**

Выполнил(а) студент(ка)

специальности 09.02.07

Информационные системы и

программирование

IV курса группы 42919/6

Леонтьев Илья

Витальевич

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

2024

Цель работы:

Ознакомление с процессом классификации на примере моделирования классов предметной области.

Ознакомление с диаграммами классов и взаимодействия UML.

**Вариант 13 (1 часть)**

Задание:

Реализовать массив данных в соответствии с вариантом задания.

Определить метод установки свойств (при недопустимых аргументах функции возвращать «false» и выдавать текст ошибки на экран).

Написать демонстрационную программу, в которой показать работу с переменными.

**Индивидуальное задание:**

Пользователь вводит массив Прямоугольников.

Свойства: две стороны.

Операции:

* Увеличение/уменьшение размера любой из сторон на определённый процент;
* Вычисление периметра и площади;
* Вычисление диагонали;

Пользователь выбирает элемент массива и выполняемую операцию.

**Вариант 13 (2 часть)**

Задание:

Перечислите несколько однотипных объектов и конкретные значение их свойств.

Выделите классы из группы однотипных объектов.

Требования:

* Не менее 5 классов;
* Не менее 7 объектов каждого класса;
* Не менее 3 свойств каждого класса;
* Не менее 3 операций для каждого класса;
* Имена классов – существительные;

**Индивидуальное задание:**

Предметная область - авиационная и космическая техника.

**Ход работы:**

Для начала я занялся первым заданием, разработал для него класс, написал его диаграмму (Рисунок 1):



Рисунок 1 – диаграмма класса Rectangle

Далее разработал код, в конце добавил в него проверки на правильность ввода.

**Код:**  
#include <iostream>

#include <vector>

#include <cmath>

#include <limits>

using namespace std;

class Rectangle {

private:

double width;

double height;

public:

// Конструктор

Rectangle(double w, double h) {

setDimensions(w, h);

}

// Метод установки размеров

bool setDimensions(double w, double h) {

if (w <= 0 || h <= 0) {

cout << "Ошибка: Размеры прямоугольника должны быть положительными." << endl;

return false;

}

width = w;

height = h;

return true;

}

// Увеличение или уменьшение размера

void adjustSize(double percent) {

width += width \* (percent / 100);

height += height \* (percent / 100);

}

// Вычисление периметра

double getPerimeter() const {

return 2 \* (width + height);

}

// Вычисление площади

double getArea() const {

return width \* height;

}

// Вычисление диагонали

double getDiagonal() const {

return sqrt(width \* width + height \* height);

}

// Метод для отображения прямоугольника

void print() const {

cout << "Прямоугольник: Ширина = " << width << ", Высота = " << height << endl;

}

};

int main() {

setlocale(LC\_ALL, "rus");

vector<Rectangle> rectangles;

int numberOfRectangles;

cout << "Введите количество прямоугольников: ";

while (true) {

try {

cin >> numberOfRectangles;

if (cin.fail() || numberOfRectangles <= 0) {

throw invalid\_argument("Введите положительное число.");

}

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear(); // очистка флага ошибки

cin.ignore(numeric\_limits<streamsize>::max(), '\n'); // игнорировать оставшийся ввод

}

}

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

double width, height;

cout << "Введите ширину и высоту прямоугольника " << (i + 1) << ": ";

while (true) {

try {

cin >> width >> height;

if (cin.fail() || width <= 0 || height <= 0) {

throw invalid\_argument("Размеры должны быть положительными числами.");

}

rectangles.emplace\_back(width, height);

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

while (true) {

cout << "\nВведите номер прямоугольника (0 для выхода): ";

int index;

while (true) {

try {

cin >> index;

if (cin.fail() || index < 0) {

throw invalid\_argument("Введите корректный номер.");

}

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

if (index == 0) break;

if (index < 1 || index > rectangles.size()) {

cout << "Неверный номер прямоугольника." << endl;

continue;

}

Rectangle& rect = rectangles[index - 1];

cout << "Выбранный прямоугольник: ";

rect.print();

cout << "Выберите операцию:\n1. Увеличить размер\n2. Уменьшить размер\n3. Вычислить периметр\n4. Вычислить площадь\n5. Вычислить диагональ\n";

int operation;

while (true) {

try {

cin >> operation;

if (cin.fail() || operation < 1 || operation > 5) {

throw invalid\_argument("Выберите операцию от 1 до 5.");

}

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

switch (operation) {

case 1: {

double percent;

cout << "На сколько процентов увеличить размеры? ";

while (true) {

try {

cin >> percent;

if (cin.fail()) {

throw invalid\_argument("Введите корректное число.");

}

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

rect.adjustSize(percent);

break;

}

case 2: {

double percent;

cout << "На сколько процентов уменьшить размеры? ";

while (true) {

try {

cin >> percent;

if (cin.fail()) {

throw invalid\_argument("Введите корректное число.");

}

break;

}

catch (const exception& e) {

cout << "Ошибка: " << e.what() << endl;

cout << "Попробуйте снова: ";

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

rect.adjustSize(-percent);

break;

}

case 3:

cout << "Периметр: " << rect.getPerimeter() << endl;

break;

case 4:

cout << "Площадь: " << rect.getArea() << endl;

break;

case 5:

cout << "Диагональ: " << rect.getDiagonal() << endl;

break;

default:

cout << "Неверная операция." << endl;

}

}

return 0;

}

Далее я выполнил второе задание: