МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**Лабораторная работа № 1:**

"Классы и объекты. Инкапсуляция"

Выполнила: студентка гр.РИС-23-3Б

Мокроусова Ангелина Андреевна

Проверила: доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2024

**Постановка задачи:**

Общая часть:

* Реализовать заданный в варианте класс
* Реализовать следующие методы: Init – метод инициализации(контроль корректности вводимых данных), Read – метод для ввода данных клавиатуры, Show – метод для вывода объекта класса в консоль
* Реализовать make – внешняя функция для создания нового экземпляра класса

Задание варианта:

Поле first – положительное целое число, целая часть числа; поле second – положительное дробное число, дробная часть числа. Реализовать метод multipy(double k) – умножение на вещественное число k

**Анализ задачи:**

1. Создаем класс lab, содержащий два приватных поля first и second, и публичные методы Init, Read, Show и multiply.
2. Метод Init будет считывать значения двух элементов с клавиатуры с помощью метода Read, проверять их корректность и в случае некорректного ввода запрашивать ввод повторно.
3. Метод Read будет задавать значения для полей first и second.
4. Метод Show будет выводить значения полей first и second на экран.
5. Метод multiply будет умножать значения полей first и second на заданное число k.
6. Функция make будет создавать объект класса lab, вызывать методы Init, Show, multiply и снова Show.
7. В функции main будет запрашиваться ввод вещественного числа для умножения и вызываться функция make.

**Код на С++ с комментариями:**

#include<iostream>

using namespace std;

class lab {

double first;

double second;

public:

void Init() {

bool flag;

do {

Read();

flag = false;

if (((first >= 0) && (second < 1) && (second > 0)) == 0) {

cout << "Элемент введен не корректно, попробуй еще раз.." << endl;

flag = true;

}

} while (flag);

}

void Read() { // метод ввода данных

cout << "Введите первый элемент(целочисленный): ";

cin >> first;

cout << "Введите второй элемент(вещественный, дробная часть): ";

cin >> second;

}

void Show() { // метод вывода данных

cout << "Первый элемент: " << first << "\tВторой элемент: " << second << endl;

}

void multiply(double k) {// метод умножения на число

first = first\*k;

second = second \* k;

}

};

void make(double k) {

lab lab1;

lab1.Init();

lab1.Show();

lab1.multiply(k);

lab1.Show();

}

int main() {

setlocale(LC\_ALL, "ru");

double k;

cout << "Введите вещественное число для умножения: ";

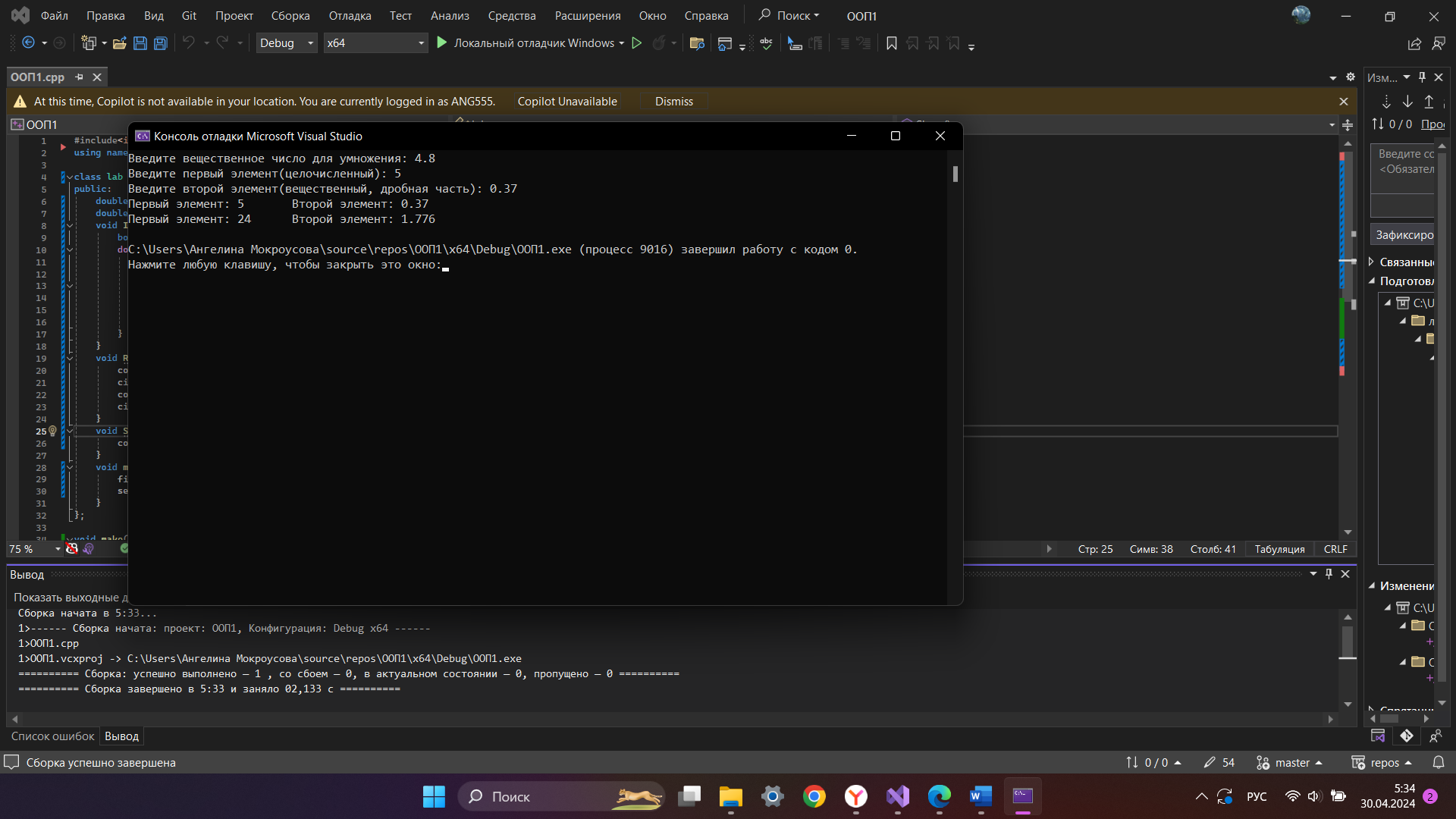
cin >> k;

make(k); // вызов функции make

return 0;

}

**Скриншот работы программы:**



**Вывод:**

Программа успешно выполняет поставленную задачу. В процессе написания был повторен и закреплен на практике пройденный ранее материал.

**Ответы на вопросы:**

1. Класс представляет из себя пользовательский тип данных. Собой он описывает абстрактную модель для некоторой группы объектов. Например, параметры “вес”, “имя” и “возраст” справедливы для описания человека. Класс, содержащий такие параметры, будет считаться абстрактной моделью какого либо человека;

2. Объект класса – конкретный предмет, который описан с помощью абстрактной модели, которую описывает класс;

3. Поля класса называются атрибутами класса;

4. Функции класса называются методами класса;

5. Спецификаторы доступа используются для проведния процесса инкапсуляции – “скрытия” тех методов или полей класса, которые не должны быть доступны его пользователю;

6. Спецификатор public используется для описания интерфейса класса – тех полей и методов, который должны быть доступны пользователю (пример – метод добавки элемента в список);

7. Спецификатор private используется для описания тех полей и методов класса, которые пользователь трогать и видеть не должен. Такие поля и методы необходимы для обеспечения внутреннего функционинрования класса, при этом им нет места в его интерфесе

8. Если описание класса начать с class, то все поля и методы в нем по умолчанию будут private;

9. Если описание класса начать с struct, то все поля и методы в нем по умолчанию будут public;

10. Для описания интерфейса класса следует использовать public, иначе пользователь банально не будет иметь доступ к интерфейсу;

11. Поля (атрибуты) класса могут быть изменны следующими образами: a) прямое обращение по форме <имя\_класса>.<поле\_класса>, если поле является публичным, б) через использования конструкторов, если конструкторы с параметрами предусмотрены, в) через функции, которые изменяют поля класса, если такие были предусмотренны;

12. Значения атрибутов класса можно получить: a) прямое обращение по форме <имя\_класса>.<поле\_класса>, если поле является публичным, б) через функции, которые возвращают значения полей класса, если такие были предусмотренны.

**Скриншот с GitHab:**