Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №18.1**

Дисциплина: «Основы теории алгоритмов и структуры данных»

Тема: Классы и объекты. Инкапсуляция.

Вариант 14

Выполнил:

студент группы РИС-20-2б

Вичугов Алексей Дмитриевич

Проверила:

доцент кафедры ИТАС

Полякова О. А.

Пермь, 2021

**Цель работы**

1. Создание консольного приложения, состоящего из нескольких файлов.
2. Использование классов и объектов в ОО программе.

**Постановка задачи**

1. Реализовать определение нового класса. Для демонстрации работы с объектами написать главную функцию. Продемонстрировать разные способы создания объектов и массивов объектов.
2. Структура-пара – структура с двумя полями, которые обычно имеют имена first и second. Требуется реализовать тип данных с помощью такой структуры. Во всех заданиях должны присутствовать:  
   а) Метод инициализации Init;  
   б) Ввод с клавиатуры Read;  
   в) Вывод на экран Show.
3. Реализовать внешнюю функцию make\_тип(), где тип – тип реализуемой структуры. Функция должна получать значения для полей структуры как параметры функции и возвращать структуру как результат.
4. Задание варианта: Поле first – положительное целое число, целая часть числа, поле second – положительное дробное число, дробная часть числа. Реализовать метод multiply(int k) – умножение на целое число k.

**Анализ задачи**

1. Определение класса:

#pragma once

struct Para{

int first;

float secnd;

public:

void Init(int,float);

void Read();

void Show();

void Calc(int k);

};

1. Определение компонентных функций:  
   #include "LabaClass1.h"

#include <iostream>

using namespace std;

void Para::Init(int f, float s){

first=f;

secnd=s;

}

void Para::Read(){

int f=0;

float s=0.0;

cout << "First (integer): ";

cin >> f;

while (cin.fail()){

cin.clear();

cin.ignore(10,'\n');

cout << "Incorrect input. Repeat, please: " << endl;

cin >> f;

}

cin.ignore(10,'\n');

cout << "Second (float): ";

cin >> s;

while (cin.fail()){

cin.clear();

cin.ignore(10,'\n');

cout << "Incorrect input. Repeat< please: " << endl;

cin >> s;

}

cout << endl;

Init (f,s);

}

void Para::Show(){cout << "Number: " << first+secnd << endl;}

void Para::Calc(int k){

float third=first+secnd;

cout << "Result: " << third\*k << endl;

}

1. Определение функции make:

Para Maker(int c, float d){

Para a;

a.Init(c,d);

return a;

}

1. Функция main():

main(){

Para b = Maker(2,0.5);

b.Show();

Para p;

p.Read();

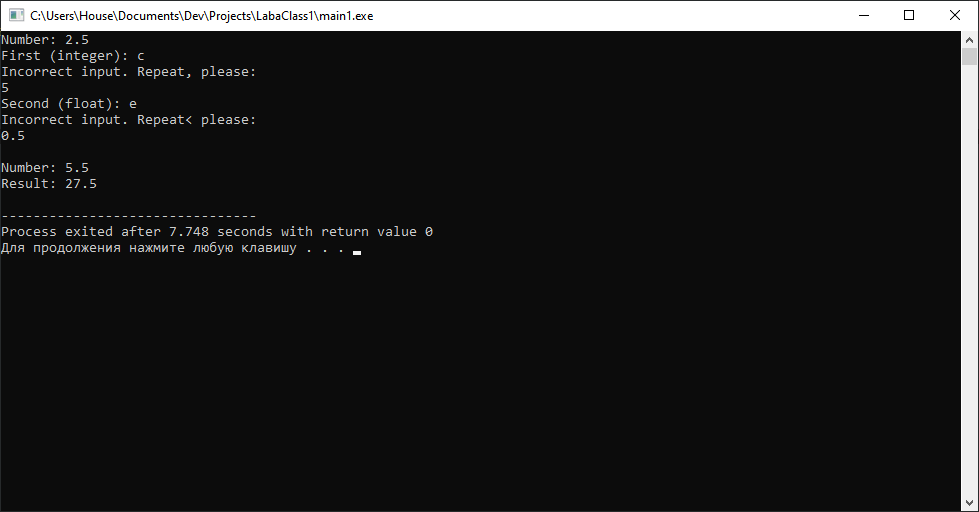
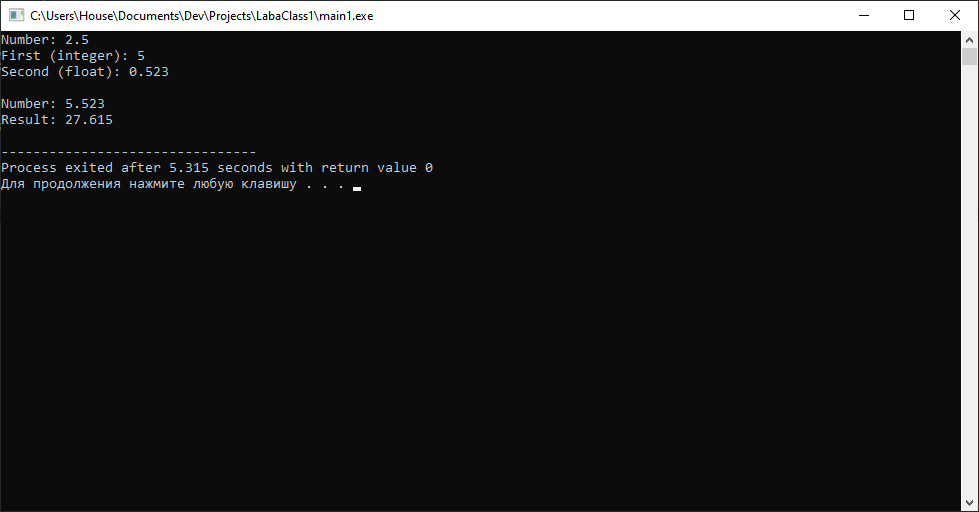
p.Show();

p.Calc(5);

return 0;

}

**Результат работы программы**



**Ответы на вопросы**

1. Класс – это конструкция, объединяющая данные и функции, которые в основном обрабатывают эти данные.
2. Объект (экземпляр класса) – это переменная класса.
3. Поля класса называются атрибутами класса.
4. Функции класса называются методами.
5. Спецификаторы доступа используются для указания возможности доступа к данным (это применяется с целью сокрытия данных, для того, чтобы уберечь данные от несанкционированного доступа или случайного изменения).
6. Спецификатор public используется для открытия доступа к полям и методам экземпляра данного класса.
7. Спецификатор private используется для сокрытия полей и методов экземпляра класса.
8. Для классов умолчанию используется спецификатор доступа private.
9. Для структур по умолчанию используется спецификатор доступа public.
10. Для описания интерфейса класса должен использоваться спецификатор public, чтобы пользователь имел возможность взаимодействовать с методами класса.
11. Изменить значения атрибутов класса можно с помощью методов класса, называемых модификаторами.
12. Получить значения атрибутов класса можно с помощью методов класса, называемых селекторами.
13. s->name;
14. s.name;
15. С помощью функции-модификатора, если она имеется.
16. С помощью функции-модификатора, если она имеется.
17. s->name;