Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе № 18.1**

Дисциплина: «Информатика»

Тема: Объектно-ориентированное программирование. Инкапсуляция

Вариант 11

Выполнил:

Студент группы РИС-20-1б

Ознобихин Елисей Андреевич

Проверила:

Доцент кафедры ИТАС

Полякова О. А.

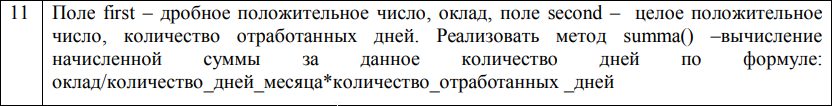
**Пермь, 2021**

**Цель работы**

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

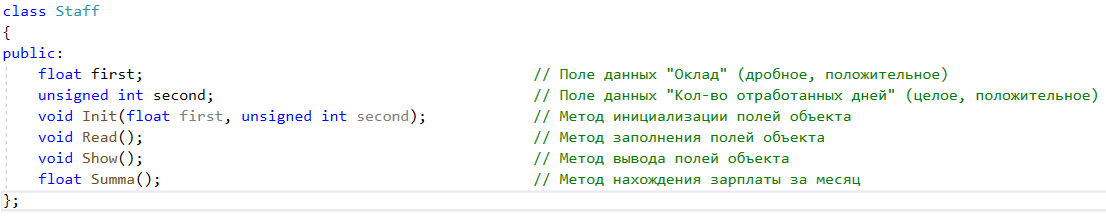
**Постановка задачи**

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:
   1. Метод инициализации Init (метод должен контролировать значения аргументов на корректность);
   2. Ввод с клавиатуры Read;
   3. 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. При передаче ошибочных параметров следует выводить сообщение и заканчивать работу.

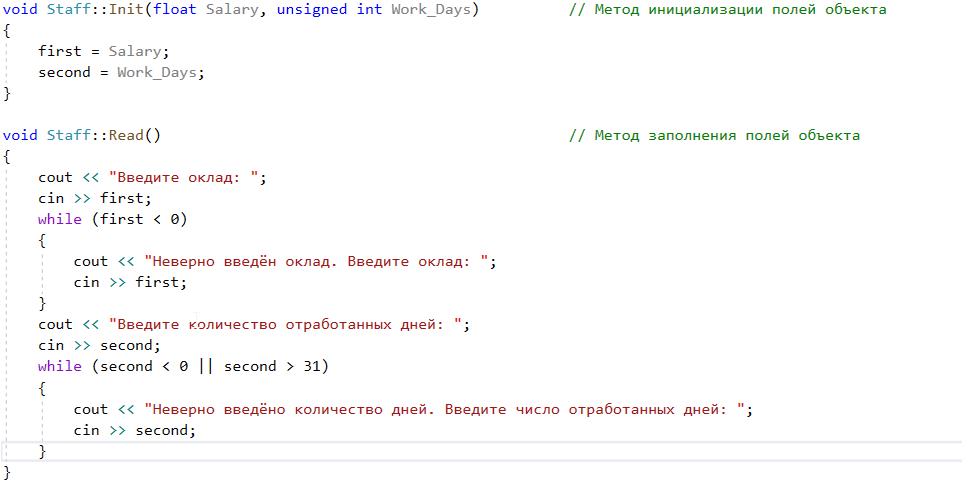


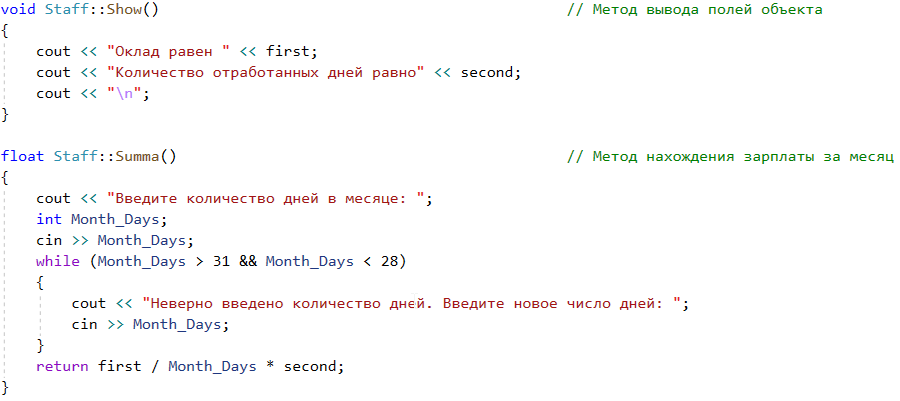
**Анализ задачи**

1. Для решения задачи необходимо:
   1. Реализовать определение класса Staff.



* 1. Реализовать определение методов класса Staff: Init, Read, Show, Summa.

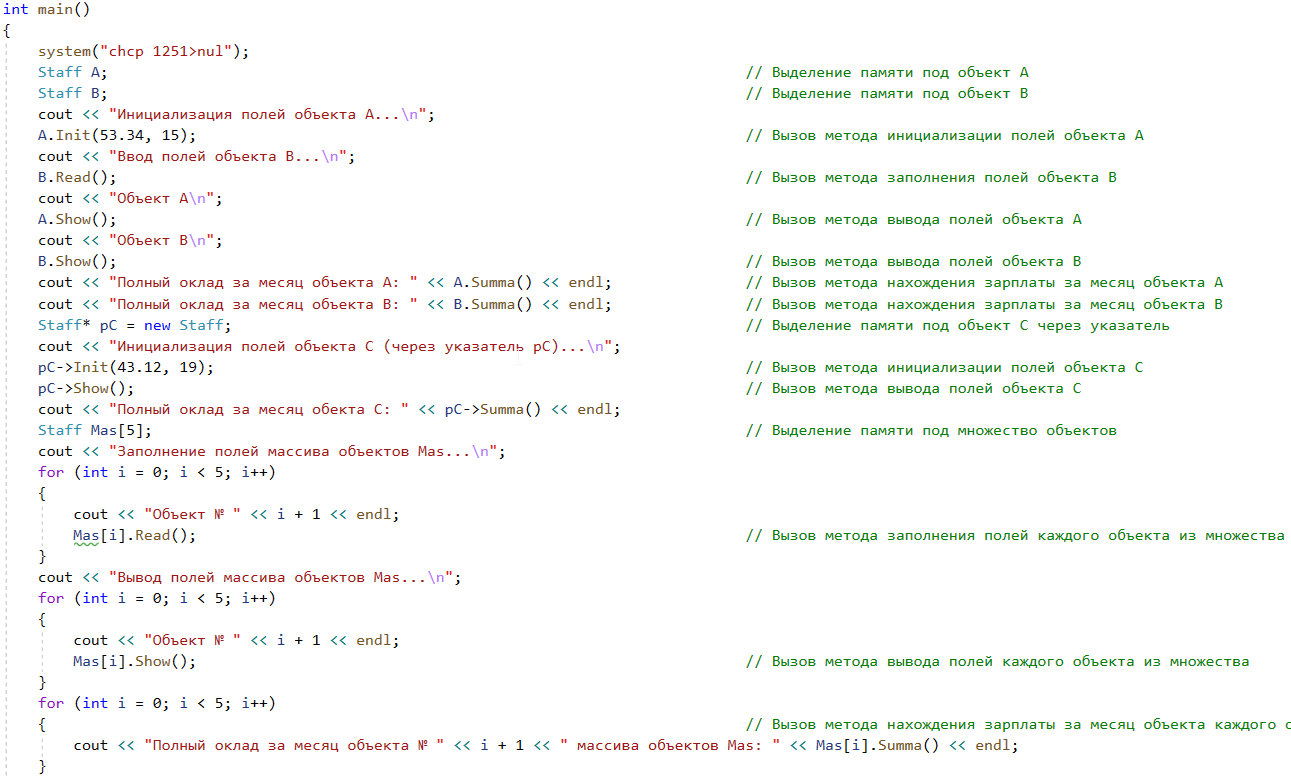


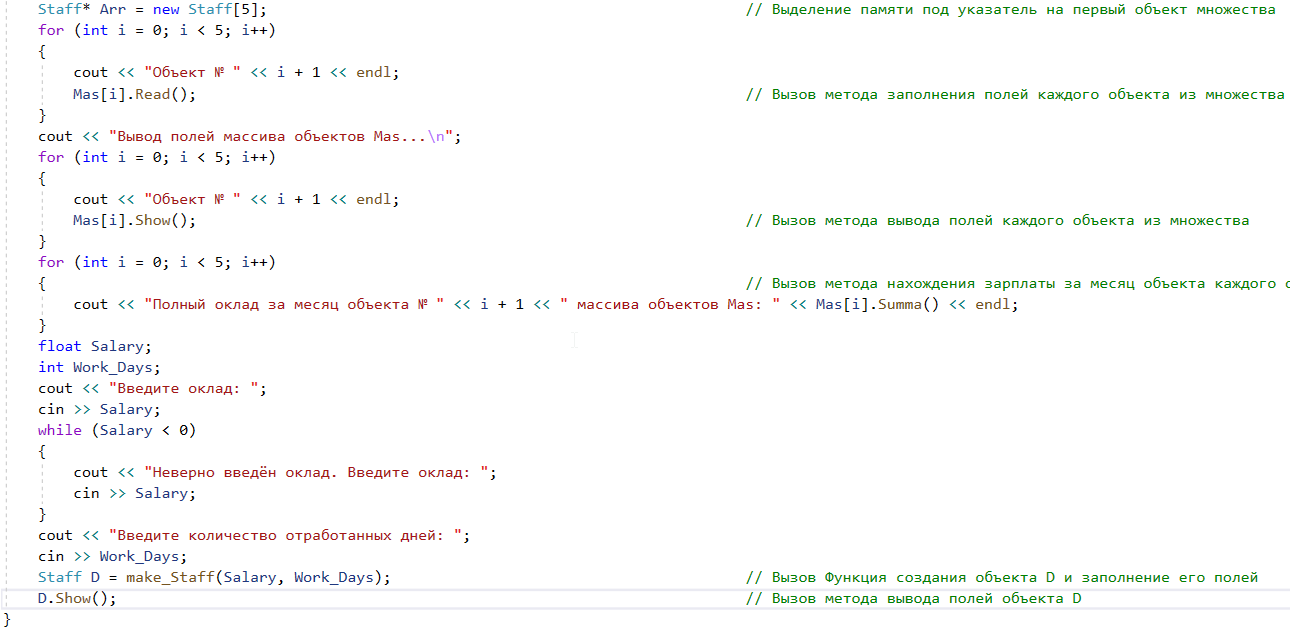


* 1. Разработать функцию make\_staff для выделения памяти под новый объект и его инициализации.



* 1. Реализовать применение этих функций в главной функции.





1. В ходе работы были использованы типы данных:
   1. Для метода Init класса Staff используются следующие аргументы:
   2. Тип float: зарплата сотрудника.
   3. Тип int: количество отработанных дней.

Сам метод имеет тип void, поскольку при работе метода не нужно возвращать значение.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_Z8VwAUEc7r.png

* 1. Для метода Read класса Staff не используются аргументы:

Сам метод имеет тип void, поскольку при работе метода не нужно возвращать значение.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_DFJemkcUMm.png

* 1. Для метода Show класса Staff не используются аргументы:

Сама метод имеет тип void, поскольку при работе метода не нужно возвращать значение.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_hQxMVSlZ2e.png

* 1. Для метода Summa класса Staff не используются аргументы:

Сам метод имеет тип void, поскольку при работе метода не нужно возвращать значение.

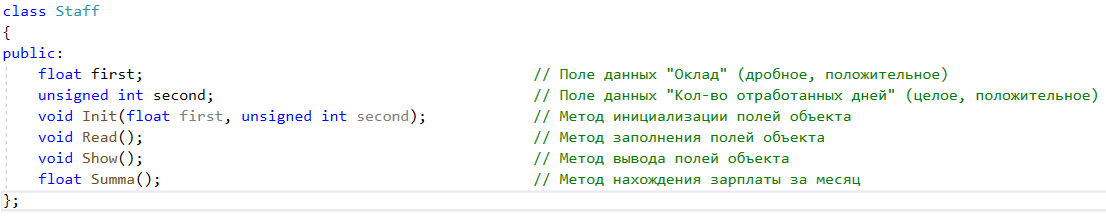
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_oTguy9ANRe.png

* 1. Для функции make\_Staff используются следующие аргументы:
  2. Тип float: зарплата сотрудника.
  3. Тип unsigned int: количество отработанных дней.

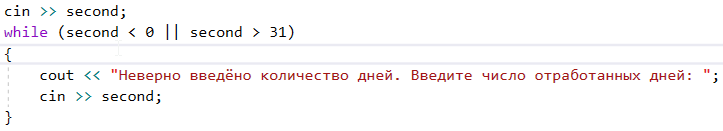
Сама функция имеет тип void, поскольку при работе функции не нужно возвращать значение.

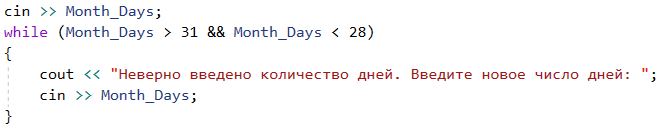
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_vcSiU1SGM8.png

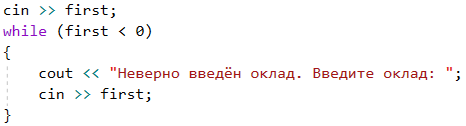
1. Для решения задачи данные были представлены в следующем виде:
   1. Для работы с данными используются атрибуты класса.



1. Для операций ввода и вывода использовались следующие операторы и функции:
   1. Ввод данных реализован с помощью оператора cin, используемого при реализации метода Read и в главной функции.







* 1. Вывод данных реализован с помощью оператора cout, используемого в главной функции и при реализации метода Show.

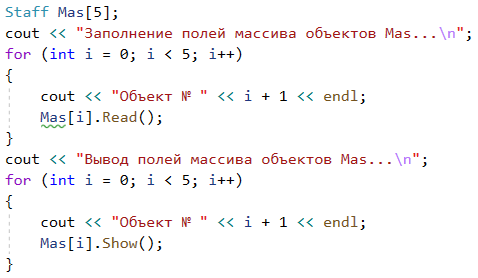
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_e1ct617d8b.png

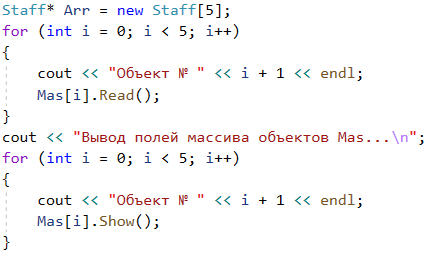
1. Поставленные задачи решены следующими действиями:
   1. Определение класса Staff было реализовано в заголовочном файле Staff.h, определение методов класса – в файле Staff.cpp. Основной блок программы и функция make\_Staff описаны в главном файле Лаба\_18\_1\_main.cpp.
   2. При работе с объектами класса Staff обращение к методам аналогичен обращению к полям структуры.

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_GpGyLK28jk.png C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_x3GcKgZnqA.png

C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_mMiVPFRCMK.png C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_VoR1OK0IqV.png

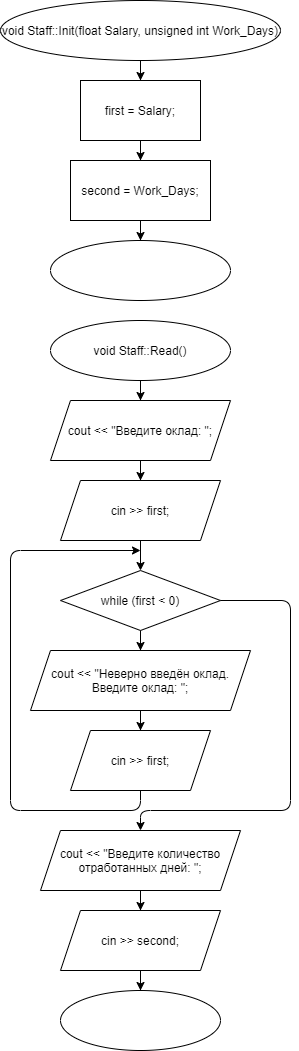
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_ibXTIbAY1L.png C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_lyrTk77tvF.png

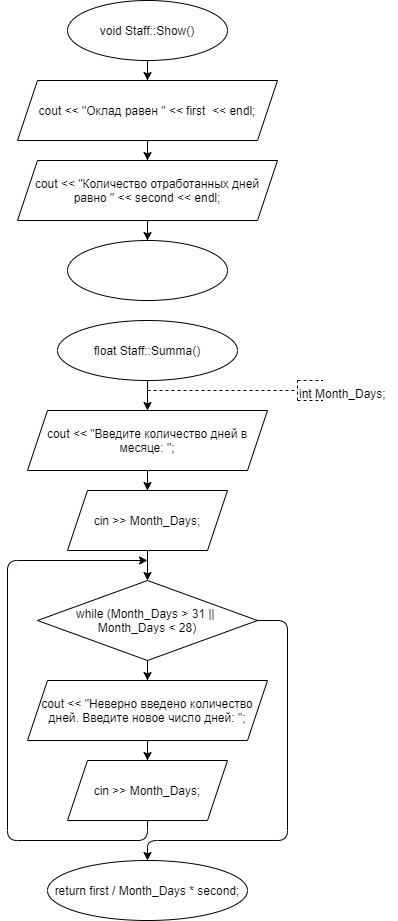


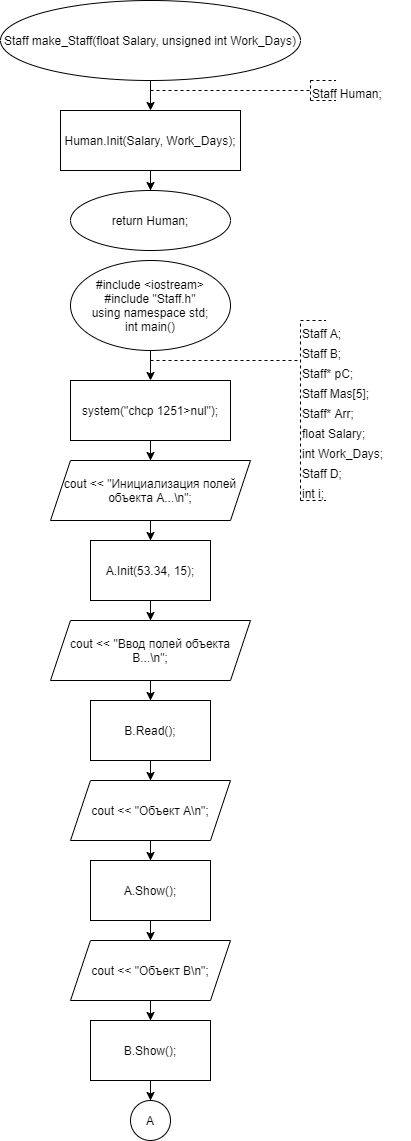


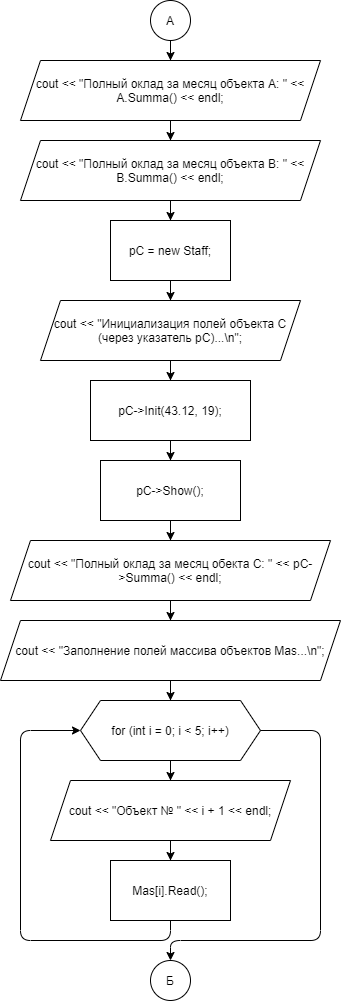
C:\Users\Асус\Documents\ShareX\Screenshots\2021-03\devenv_jmfLQx5eyQ.png

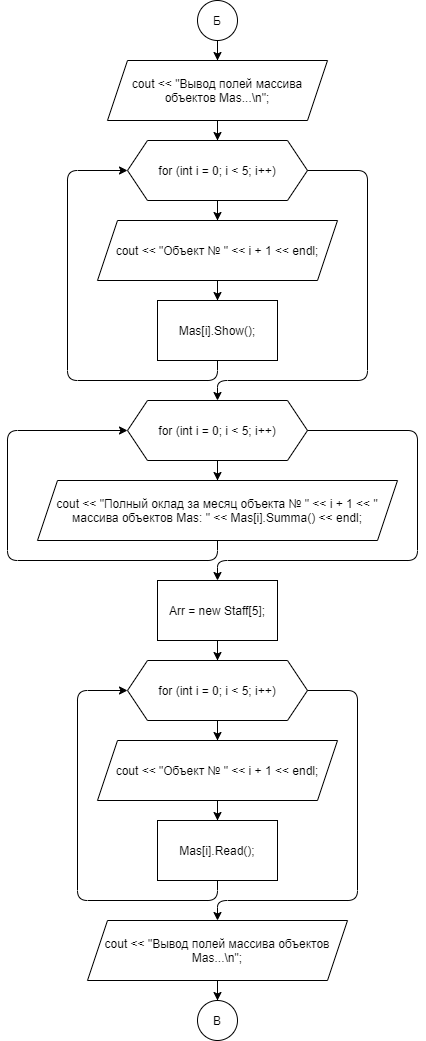
**Блок-схема программы**

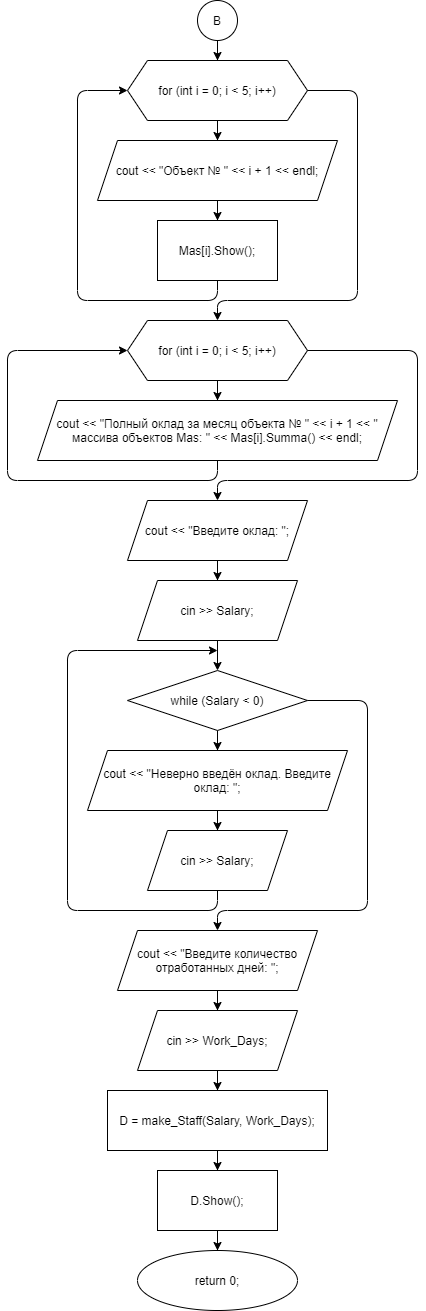












**Реализация задачи на языке С++**

**Заголовочный файл staff.h**

#pragma once

class Staff

{

public:

float first;

unsigned int second;

void Init(float first, unsigned int second);

void Read();

void Show();

float Summa();

};

**Файл с описанием методов класса staff.cpp**

#include <iostream>

#include "Staff.h"

using namespace std;

void Staff::Init(float Salary, unsigned int Work\_Days)

{

first = Salary;

second = Work\_Days;

}

void Staff::Read()

{

cout << "Введите оклад: ";

cin >> first;

while (first < 0)

{

cout << "Неверно введён оклад. Введите оклад: ";

cin >> first;

}

cout << "Введите количество отработанных дней: ";

cin >> second;

while (second < 0 || second > 31)

{

cout << "Неверно введёно количество дней. Введите число отработанных дней: ";

cin >> second;

}

}

void Staff::Show()

{

cout << "Оклад равен " << first << endl;

cout << "Количество отработанных дней равно " << second;

cout << "\n";

}

float Staff::Summa()

{

cout << "Введите количество дней в месяце: ";

int Month\_Days;

cin >> Month\_Days;

while (Month\_Days > 31 || Month\_Days < 28)

{

cout << "Неверно введено количество дней. Введите новое число дней: ";

cin >> Month\_Days;

}

return first / Month\_Days \* second;

}

**Файл с главной программой Лаба\_18\_1\_main.cpp**

#include <iostream>

#include "Staff.h"

using namespace std;

Staff make\_Staff(float Salary, unsigned int Work\_Days)

{

Staff Human;

Human.Init(Salary, Work\_Days);

return Human;

}

int main()

{

system("chcp 1251>nul");

Staff A;

Staff B;

cout << "Инициализация полей объекта A...\n";

A.Init(53.34, 15);

cout << "Ввод полей объекта B...\n";

B.Read();

cout << "Объект A\n";

A.Show();

cout << "Объект B\n";

B.Show();

cout << "Полный оклад за месяц объекта A: " << A.Summa() << endl;

cout << "Полный оклад за месяц объекта B: " << B.Summa() << endl;

Staff\* pC = new Staff;

cout << "Инициализация полей объекта C (через указатель pC)...\n";

pC->Init(43.12, 19);

pC->Show();

cout << "Полный оклад за месяц обекта C: " << pC->Summa() << endl;

Staff Mas[5];

cout << "Заполнение полей массива объектов Mas...\n";

for (int i = 0; i < 5; i++)

{

cout << "Объект № " << i + 1 << endl;

Mas[i].Read();

}

cout << "Вывод полей массива объектов Mas...\n";

for (int i = 0; i < 5; i++)

{

cout << "Объект № " << i + 1 << endl;

Mas[i].Show();

}

for (int i = 0; i < 5; i++)

cout << "Полный оклад за месяц объекта № " << i + 1 << " массива объектов Mas: " << Mas[i].Summa() << endl;

}

Staff\* Arr = new Staff[5];

for (int i = 0; i < 5; i++)

{

cout << "Объект № " << i + 1 << endl;

Mas[i].Read();

}

cout << "Вывод полей массива объектов Mas...\n";

for (int i = 0; i < 5; i++)

{

cout << "Объект № " << i + 1 << endl;

Mas[i].Show();

}

for (int i = 0; i < 5; i++)

cout << "Полный оклад за месяц объекта № " << i + 1 << " массива объектов Mas: " << Mas[i].Summa() << endl;

}

float Salary;

int Work\_Days;

cout << "Введите оклад: ";

cin >> Salary;

while (Salary < 0)

{

cout << "Неверно введён оклад. Введите оклад: ";

cin >> Salary;

}

cout << "Введите количество отработанных дней: ";

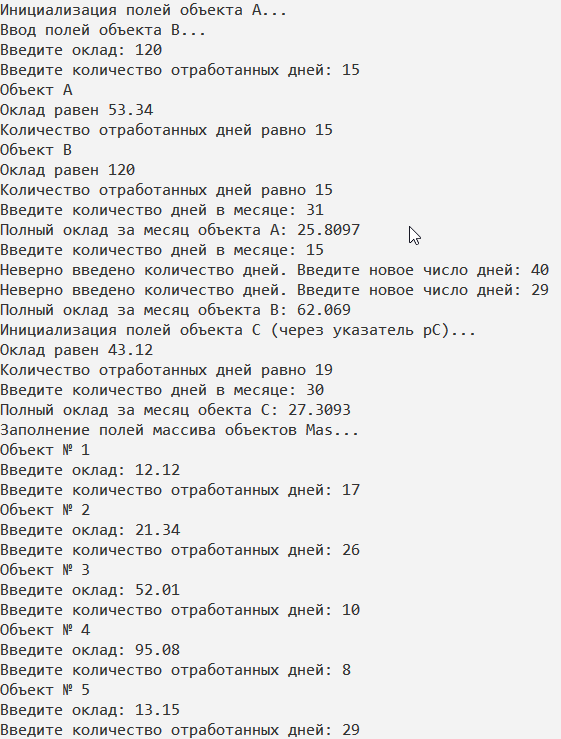
cin >> Work\_Days;

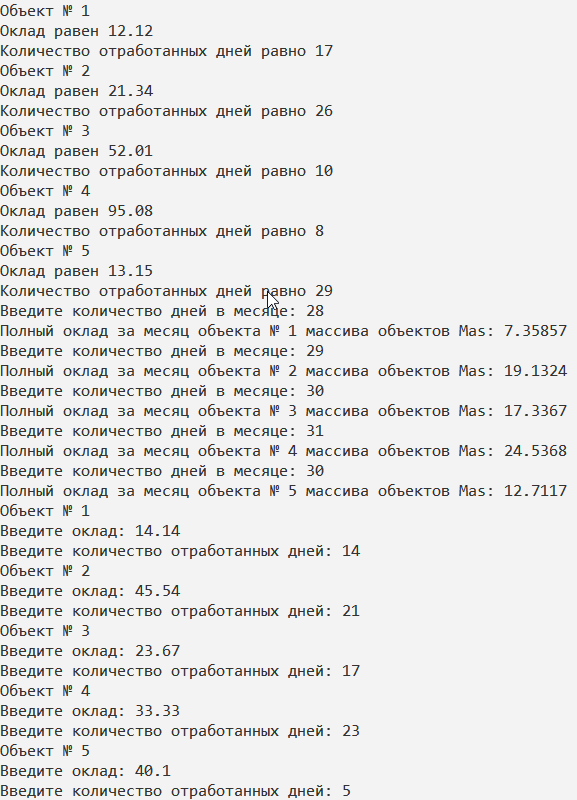
Staff D = make\_Staff(Salary, Work\_Days);

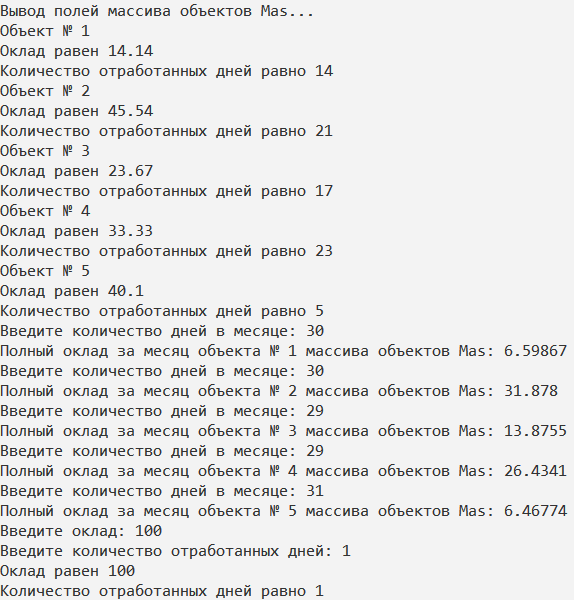
D.Show();

}

**Скриншоты результатов работы программы**







**Ответы на вопросы**

1. Что такое класс?

Класс – абстрактный тип данных, определяемый пользователем, представляет собой модель реального объекта в виде дынных и функций для работы с ними.

1. Что такое объект (экземпляр) класса?

Любая переменная ранее созданного класса называется объектом или экземпляром класса.

1. Как называются поля класса?

Поля в классе называются атрибутами.

1. Как называются функции класса?

Функции в классе называются методами.

1. Для чего используются спецификаторы доступа?

Спецификаторы доступа используются для управления видимости элементов класса (атрибутов и методов).

1. Для чего используется спецификатор public?

Спецификатор public используется для возможности взаимодействия с элементами с данным спецификатором и видимости этих элементов вне класса (обычно это методы).

1. Для чего используется спецификатор private?

Спецификатор private используется для видимости элементов только внутри класса (обычно это атрибуты класса).

1. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использован по умолчанию? (private)

Для класса, начинающегося со спецификатора class, по умолчанию используется спецификатор доступа private.

1. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использован по умолчанию?

Для класса, начинающегося со спецификатора struct, по умолчанию используется спецификатор доступа public.

1. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

При описании интерфейса используется спецификатор public, поскольку данный спецификатор обеспечивает возможность использования описанных в классе методов вне данного класса.

1. Каким образом можно изменить значения атрибутов экземпляра класса?

Для изменения атрибутов экземпляра класса обычно используют определенными в классе специальными методами, называемыми модификаторами.

1. Каким образом можно получить значения атрибутов экземпляра класса?

Для получения атрибутов экземпляра класса обычно используют определенными в классе специальными методами, называемыми селекторами.

1. Класс описан следующим образом

struct Student

{

string name:

int group;

……….

};

Объект класса определен следующим образом

Student\* s = new Student;

Как можно обратиться к полю name объекта s?

Обращение к полю name объекта s аналогично обращению к полям указателя структуры и выглядит следующим образом:

s->name;

1. Класс описан следующим образом

struct Student

{

string name:

int group;

……….

};

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

Обращение к полю name объекта s аналогично обращению к полям переменной структуры и выглядит следующим образом:

s.name;

1. Класс описан следующим образом

class Student

{

string name:

int group;

……….

};

Объект класса определен следующим образом

Student\* s = new Student;

Как можно обратиться к полю name объекта s?

Обращение к полю name объекта s в данном случае невозможно, т. к. все элементы класса имеют спецификатор доступа private.

1. Класс описан следующим образом

class Student

{

string name:

int group;

public:

……….

};

Объект класса определен следующим образом

Student s;

Как можно обратиться к полю name объекта s?

Обращение к полю name объекта s возможно через селекторы и модификаторы класса.

1. Класс описан следующим образом

class Student

{

public:

char\* name:

int group;

……….

};

Объект класса определен следующим образом

Student\* s = new Student;

Как можно обратиться к полю name объекта s?

Обращение к полю name объекта s возможно через селекторы, модификаторы и аналогично обращению к полям указателя структуры.