Пермский национальный исследовательский политехнический университет

**Лабораторная работа №2**

**Дисциплина: Информатика**

**Вариант № 9**

**"Классы и объекты. Использование конструкторов"**

Выполнил: Зайченко Никита Геннадьевич

Проверила: Доцент кафедры ИТАС Полякова О. А.

Пермь 2022

**Содержание отчета**

1) Постановка задачи (общая и конкретного варианта).

2) Описание класса.

3) Определение компонентных функций.

4) Определение функций make() и print().

5) Объяснение результатов работы программы.

**Постановка задачи (общая и конкретного варианта)**

1) Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.

2) Создание объектов с использованием конструкторов.

3) Пользовательский класс СОТРУДНИК

ФИО – string; Должность – string; Зарплата – double;

**Описание класса.**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Employee

{

//атрибуты

string name;

string post;

double zarplata;

public:

Employee();//конструктор без параметров

Employee(string, string, double);//конструктор с параметрами

Employee(const Employee&);//конструктор копирования

~Employee();//деструктор

string get\_name();//селектор

void set\_name(string);//модификатор

string get\_post();//селектор

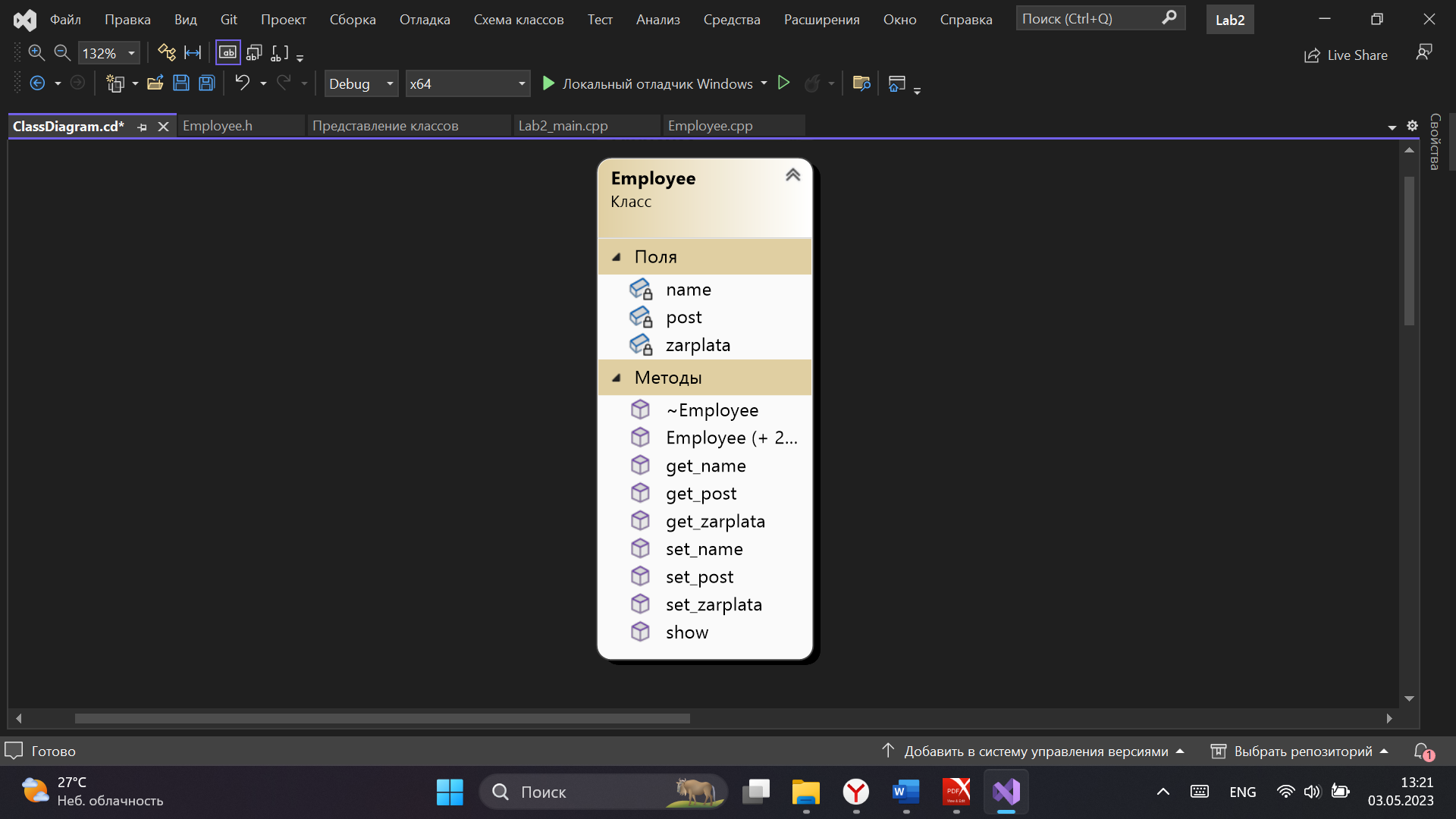
void set\_post(string); //модификатор

double get\_zarplata();//селектор

void set\_zarplata(double); //модификатор

void show();//просмотр атрибутов

};



**Определение компонентных функций.**

//конструктор без параметров

Employee::Employee()

{

name = "";

post = "";

zarplata = 0;

cout << endl << "Конструктор без параметров для обьекта: " << this << endl;

}

//конструктор с параметрами

Employee::Employee(string Name, string Post, double Zarplata)

{

name = Name;

post = Post;

zarplata = Zarplata;

cout << endl << "Конструктор с параметром для обьекта: " << this << endl;

}

//конструктор копирования

Employee::Employee(const Employee& t)

{

name = t.name;

post = t.post;

zarplata = t.zarplata;

cout << endl << "Конструктор копирования для обьекта: " << this << endl;

}

//деструктор

Employee::~Employee()

{

cout << endl << "Деструктор для обьекта: " << this << endl;

}

//селекторы

string Employee::get\_name()

{

return name;

}

string Employee::get\_post()

{

return post;

}

double Employee::get\_zarplata()

{

return zarplata;

}

//модификаторы

void Employee::set\_name(string newName)

{

name = newName;

}

void Employee::set\_post(string newPost)

{

post = newPost;

}

void Employee::set\_zarplata(double newZarplata)

{

zarplata = newZarplata;

}

//метод для просмотра атрибутов

void Employee::show()

{

cout << endl;

cout << "ФИО: " << name << endl;

cout << "Должность: " << post << endl;

cout << "Зарплата: " << zarplata << endl;

}

**Определение функции make() и print().**

//функция для возврата объекта как результата

Employee make\_tovar()

{

string name;

string post;

double zarplata;

cout << endl << "Введите ФИО: ";

getline(cin, name);

cout << endl << "Введите должность: ";

getline(cin, post);

cout << endl << "Введите зароботную плату: ";

cin >> zarplata;

Employee t(name, post, zarplata);

return t;

}

//функция для передачи объекта как параметра

void print\_tovar(Employee t)

{

t.show();

}

**Объяснение результатов работы программы.**

Данный код представляет собой определение класса Employee, содержащего три атрибута: имя сотрудника (name), его должность (post) и зарплату (zarplata). Также в классе объявлены следующие функции:

- Конструкторы: конструктор без параметров, конструктор с параметрами и конструктор копирования (Employee(), Employee(string, string, double), Employee(const Employee&));

- Деструктор (~Employee());

- Селекторы и модификаторы для каждого из атрибутов (get\_name(), set\_name(string), get\_post(), set\_post(string), get\_zarplata(), set\_zarplata(double));

- Функция просмотра атрибутов класса (show());

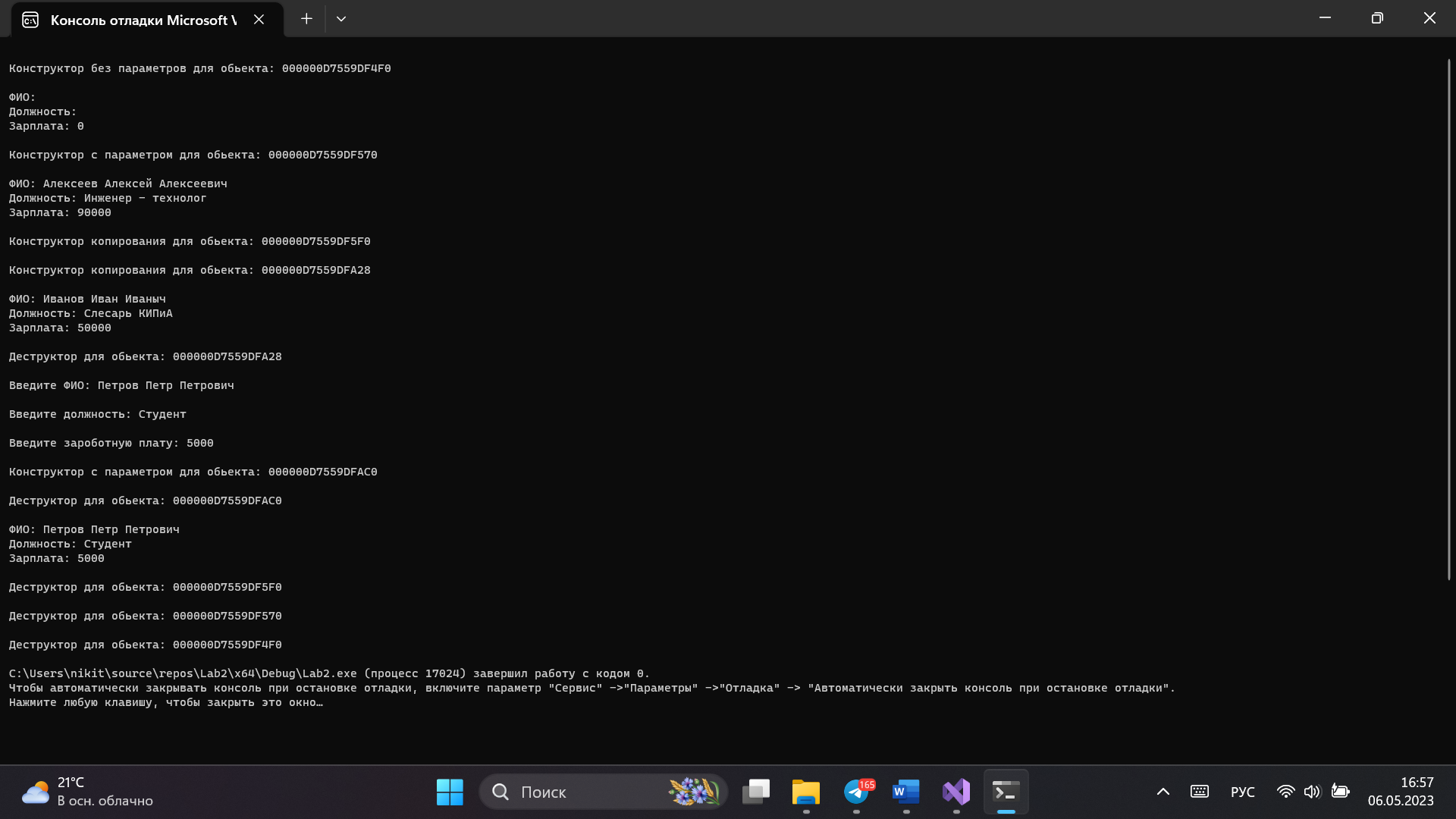
- Функция make\_tovar(), которая запрашивает у пользователя значения атрибутов и создает объект класса Employee, возвращая его;

- Функция print\_tovar(), которая принимает объект класса Employee в качестве параметра и вызывает его метод show().

При вызове функции make\_tovar(), пользователю выводятся приглашения на ввод ФИО, должности и зарплаты сотрудника. Затем создается объект класса Employee с заданными атрибутами и возвращается функцией.

Функция print\_tovar() принимает на вход объект класса Employee и выводит значения его атрибутов на экран, вызывая метод show().

Таким образом, эти функции используются для создания и вывода информации о сотруднике.



**Ответы на контрольные вопросы**

**1. Для чего нужен конструктор?**

Конструктор используется для создания объектов класса и их инициализации значениями по умолчанию или заданными параметрами.

**2. Сколько типов конструкторов существует в С++?**

В С++ существует три типа конструкторов: конструктор без параметров, конструктор с параметрами и конструктор копирования.

**3. Для чего используется деструктор? В каких случаях деструктор описывается явно?**

Деструктор используется для освобождения ресурсов, выделенных для объекта класса. Деструктор описывается явно, если объект класса использует динамическую память или другие ресурсы, которые требуют освобождения.

**4. Для чего используется конструктор без параметров? Конструктор с параметрами? Конструктор копирования?**

Конструктор без параметров используется для создания объекта с значениями по умолчанию. Конструктор с параметрами используется для создания объекта с заданными значениями. Конструктор копирования используется для создания объекта-копии.

**5. В каких случаях вызывается конструктор копирования?**

Конструктор копирования вызывается при создании копии объекта, например, при передаче объекта по значению в функцию или при возвращении объекта из функции.

**6. Перечислить свойства конструкторов.**

Свойства конструкторов: имя должно совпадать с именем класса, не возвращают значения, могут быть перегружены, вызываются автоматически при создании объекта, могут иметь различное число и тип параметров.

**7. Перечислить свойства деструкторов.**

Свойства деструкторов: имя начинается с символа ~, не имеют параметров, не возвращают значения, вызываются автоматически при удалении объекта.

**8. К каким атрибутам имеют доступ методы класса?**

Методы класса имеют доступ к атрибутам объекта класса и к другим методам класса.

**9. Что представляет собой указатель this?**

Указатель this представляет адрес объекта класса, для которого вызывается метод.

**10. Какая разница между методами определенными внутри класса и вне класса?**

Методы, определенные внутри класса, являются методами-членами класса и имеют доступ к его атрибутам и методам. Методы, определенные вне класса, называются методами-друзьями и имеют доступ к приватным атрибутам и методам класса.

**11. Какое значение возвращает конструктор?**

Конструктор не возвращает значения.

**12. Какие методы создаются по умолчанию?**

По умолчанию создаются конструктор без параметров, деструктор, конструктор копирования, конструктор с параметрами, оператор присваивания, оператор равенства, оператор неравенства, оператор меньше, оператор больше, оператор меньше или равно, оператор больше или равно.

**13. Какое значение возвращает деструктор?**

Деструктор не возвращает значения.

**14. Дано описание класса**

**class Student**

**{**

**string name;**

**int group;**

**public:**

**student(string, int);**

**student(const student&)**

**~student();**

**};**

**Какой метод отсутствует в описании класса?**

Нет метода show(), который используется в функции print().

**15. Какой метод будет вызван при выполнении следующих операторов:**

**student\*s;**

**s=new student;**

Будет вызван конструктор без параметров.

**16. Какой метод будет вызван при выполнении следующих операторов:**

**student s(“Ivanov”,20);**

Будет вызван конструктор с параметрами.

**17. Какие методы будут вызваны при выполнении следующих операторов:**

**student s1(“Ivanov”,20);**

**student s2=s1;**

Будет вызван конструктор копирования.

**18. Какие методы будут вызваны при выполнении следующих операторов:**

**student s1(“Ivanov”,20);**

**student s2;**

**s2=s1;**

Будет вызван оператор присваивания.

**19. Какой конструктор будет использоваться при передаче параметра в функцию print():**

**void print(student a)**

**{a.show();}**

Будет вызван конструктор копирования.

**20. Класс описан следующим образом:**

**class Student**

**{**

**string name;**

**int age;**

**public:**

**void set\_name(string);**

**void set\_age(int ); ….. };**

**Student p;**

**Каким образом можно присвоить новое значение атрибуту name объекта р?**

Для изменения значения атрибута name можно использовать метод setname(string) класса Student: p.setname("Новое имя").