Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

(ПНИПУ)

**Факультет:** электротехнический

**Направление:** Промышленная мехатроника и робототехника

ОТЧЁТ

По лабораторной работе №2

«Классы»

Семестр 1

Выполнил:

студент группы ПРТ-21-1б

Поздняков Е.С.

Пермь 2022

**Задание №1**

**Постановка задачи**

Определить пользовательский класс. Определить в классе следующие конструкторы: конструктор без параметров, конструктор с параметром, конструктор копирования. Определить в классе деструктор. Определить селекторы и модификаторы. Написать демонстрационную программу, в которой продемонстрированы все три вызова конструкторов.

Пользовательский класс ЭКЗАМЕН

ФИО студента – string

Предмет – string

Оценка - int

**Анализ**

1. Создается экземпляр chel класса Exam, поля этого класса инициализируются значениями, заданными пользователем, при помощи конструктора с параметрами.
2. При помощи конструктора без параметров выводятся другие значения полей.
3. При помощи конструктора копирования создаётся еще один экземпляр chel2. Полям этого экземпляра присваиваются значения полей chel.

**Используемые переменные**

string name – имя студента;

string lesson – название предмета;

int mark – оценка.

**Код C++**

#include <iostream>

#include <cmath>

#include <string>

using namespace std;

class Exam

{

private:

string name;

string lesson;

int mark;

public:

Exam()

{

this->name = "-";

this->lesson = "none";

this->mark = 0;

}

Exam (string name, string lesson, int mark)

{

this->name = name;

this->lesson = lesson;

this->mark = mark;

}

Exam (const Exam& p)

{

this->name = p.name;

this->lesson = p.lesson;

this->mark = p.mark;

}

void Print()

{

cout << "Name: " << name << endl

<< "Lesson: " << lesson << endl

<< "Mark: " << mark << endl;

cout << endl;

}

~Exam()

{

cout << "Destructor" << endl;

}

};

int main()

{

string name, lesson;

int mark;

cout << "Input name: ";

cin >> name;

cout << "Input lesson: ";

cin >> lesson;

cout << "Input mark: ";

cin >> mark;

Exam chel(name, lesson, mark);

chel.Print();

Exam chel1 = Exam();

chel1.Print();

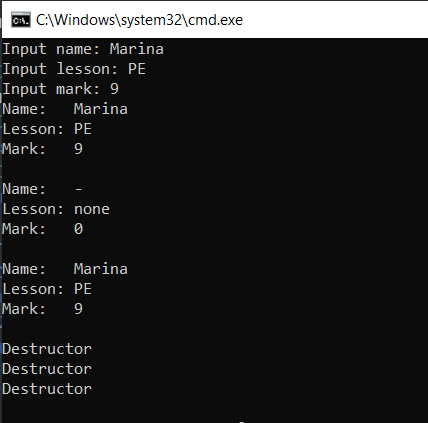
Exam chel2 = Exam(chel);

chel2.Print();

return 0;

}

**Скриншоты результатов**

****

**Анализ результатов**

Пользователь вводит значения трёх полей. Выводятся значения полей первого экземпляра. Далее выводятся значения полей, заданных по умолчанию. После этого при помощи конструктора копирования создается новый экземпляр, поля которого равны значениям полей первого экземпляра. В конце срабатывает деструктор.

**Контрольные вопросы**

1. Для чего нужен конструктор?

Конструктор нужен для инициализации полей экземпляров.

1. Сколько типов конструкторов существует в C++?

Существует три типа конструкторов: с параметрами, без параметров и копирования.

1. Для чего используется деструктор?

Деструктор используется для освобождения динамической памяти.

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

Конструктор без параметров – для инициализации полей значениями по умолчанию, с параметрами – для инициализации заданными значениями, копирования – для создания экземпляра, значения полей которого соответствуют значениям полей другого поля.

1. В каких случаях вызывается конструктор копирования?

Для создания экземпляра, значения полей которого соответствуют значениям полей другого поля

1. Перечислить свойства конструкторов.

- у конструктора не указывается возвращаемый тип

- имя конструктора совпадает с именем класса

- конструкторы можно перегружать

- конструктор располагается в public

7. Перечислить свойства деструкторов.

- деструктор ничего не принимает и ничего не возвращает

- имя деструктора такое же, как у класса, но со значком ~ перед именем.

- деструктор располагается в public

- деструктор может быть только один

8. К каким атрибутам имеют доступ методы класса?

Методы класса имеют доступ к атрибутам из private

9. Что представляет собой указатель this?

Указатель this – указатель на поле класса.

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

Методы класса могут взаимодействовать с полями класса, в отличие от методов, представленных вне класса.

11. Какое значение возвращает конструктор?  
Конструктор инициализирует значение полей.

12. Какие методы создаются по умолчанию?

По умолчанию создаются методы, которые ничего не принимают и не возвращают.

13. Какое значение возвращает деструктор?  
Деструктор ничего не возвращает.

14. В описании класса отсутствует конструктор без параметров.

15. Какой метод будет вызван при выполнении следующих операторов:

student\* s;

s = new student;

Конструктор копирования.

16. Какой метод будет вызван при выполнении следующих операторов:

Student s(“Ivanov”, 20)

Конструктор с параметрами

17. Какие методы будут вызваны при выполнении следующих операторов:

Student s1(“Ivanov”, 20)

Student s2 = s1;

Конструктор с параметрами и копирования

18. Какие методы будут вызваны при выполнении следующих операторов:

Student s1(“Ivanov”, 20)

Student s2;

s2 = s1;

Конструктор с параметрами, без параметров и копирования.

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

Void print(student a)

{

a.show();

}

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 p?

При помощи метода set\_name(string name) {this->name = name}

Int main()

{

p.set\_name(name);

}