Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: «ООТПиСП»

Тема: «КЛАССЫ И ОБЪЕКТЫ В С++»

Вариант №3

Выполнил:

Студент 2 курса

Группы ПО-7

Комиссаров А.Е.

Проверила:

Хацкевич М.В.

2022

**Цель работы:** получить практические навыки реализации классов на С++.

**Задание:**

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение).

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

3. Определить в классе деструктор.

4. Определить в классе компоненты-функции для просмотра и установки полей данных.

5. Определить указатель на компоненту-функцию.

6. Определить указатель на экземпляр класса.

7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).

8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

3. КАДРЫ

имя – char\*

номер цеха – int

разряд – int

**Код программы:**

**Файл Personel.h:**

**const int LNAME = 25;**

**#include <iostream>**

**#include <iomanip>**

**#include <string.h>**

**#include <string>**

**#include <stdio.h>**

**class Personel {**

**private:**

**char name[LNAME];**

**int fabId;**

**int level;**

**public:**

**Personel();**

**Personel(char\*, int, int);**

**Personel(const Personel & refObj);**

**~Personel();**

**char\* getName();**

**int getId();**

**int getLevel();**

**void setName(char\*);**

**void setId(int);**

**void setLevel(int);**

**void set(char\*, int, int);**

**};**

**Файл Personel.cpp:**

**#include "Personel.h"**

**Personel::Personel() {**

**std::cout << "Using a default constructor." << std::endl;**

**this->set((char\*)"Default", 0, 0);**

**}**

**Personel::Personel(char\* newName, int newId, int newLevel) {**

**std::cout << "Using a constructor with parameters." << std::endl;**

**this->set(newName, newId, newLevel);**

**}**

**Personel::Personel(const Personel& refObj) {**

**std::cout << "Using a copying constructor." << std::endl;**

**this->set((char\*)refObj.name, refObj.fabId, refObj.level);**

**}**

**Personel::~Personel() {**

**std::cout << "Executing a destructor of a Personel object." << std::endl;**

**}**

**char\* Personel::getName() {**

**return this->name;**

**}**

**int Personel::getId() {**

**return this->fabId;**

**}**

**int Personel::getLevel() {**

**return this->level;**

**}**

**void Personel::setName(char\* newName) {**

**for (int i = 0; i < 25; i++) {**

**this->name[i] = newName[i];**

**}**

**}**

**void Personel::setId(int newId) {**

**this->fabId = newId;**

**}**

**void Personel::setLevel(int newLevel) {**

**this->level = newLevel;**

**}**

**void Personel::set(char\* newName, int newId, int newLevel) {**

**this->setName(newName);**

**this->setId(newId);**

**this->setLevel(newLevel);**

**}**

**Файл Source.cpp:**

**#include "Personel.h"**

**int main() {**

**Personel pers1;**

**Personel pers2((char\*)"HelloWorld", 25, 3);**

**Personel pers3(pers2);**

**pers3.setName((char\*)"ThisIsPers3(copyOfPers2)"); //setters work**

**char\*(Personel::\*myptr)(); //pointer definition**

**myptr = &Personel::getName; //setting up a pointer**

**Personel\* obj; //object pointer example**

**obj = &pers2;**

**std::cout << "Default: " << pers1.getName() << std::endl;**

**std::cout << "With param: " << pers2.getName() << std::endl;**

**std::cout << "Copy: " << pers3.getName() << std::endl;**

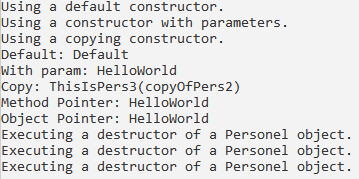
**std::cout << "Method Pointer: " << (pers2.\*myptr)() << std::endl;**

**std::cout << "Object Pointer: " << obj->getName() << std::endl;**

**return 0;**

**}**

**Результат работы программы:**

****

**Выполненные задания:**

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение).

class Personel {

private:

char name[LNAME]; int fabId; int level;

public:

Personel();

Personel(char\*, int, int);

Personel(const Personel & refObj);

~Personel();

char\* getName();

int getId();

int getLevel();

void setName(char\*);

void setId(int);

void setLevel(int);

void set(char\*, int, int);

};

2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования.

Personel::Personel() {

std::cout << "Using a default constructor." << std::endl;

this->set((char\*)"Default", 0, 0);

}

Personel::Personel(char\* newName, int newId, int newLevel) {

std::cout << "Using a constructor with parameters." << std::endl;

this->set(newName, newId, newLevel);

}

Personel::Personel(const Personel& refObj) {

std::cout << "Using a copying constructor." << std::endl;

this->set((char\*)refObj.name, refObj.fabId, refObj.level);

}

3. Определить в классе деструктор.

Personel::~Personel() {

std::cout << "Executing a destructor of a Personel object." << std::endl;

}

4. Определить в классе компоненты-функции для просмотра и установки полей данных.

char\* Personel::getName() {

return this->name;

}

int Personel::getId() {

return this->fabId;

}

int Personel::getLevel() {

return this->level;

}

void Personel::setName(char\* newName) {

for (int i = 0; i < 25; i++) {

this->name[i] = newName[i];

}

}

void Personel::setId(int newId) {

this->fabId = newId;

}

void Personel::setLevel(int newLevel) {

this->level = newLevel;

}

void Personel::set(char\* newName, int newId, int newLevel) {

this->setName(newName);

this->setId(newId);

this->setLevel(newLevel);

}

5. Определить указатель на компоненту-функцию.

char\*(Personel::\*myptr)(); //pointer definition

myptr = &Personel::getName; //setting up a pointer

6. Определить указатель на экземпляр класса.

Personel\* obj; //object pointer example

obj = &pers2;

7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал).

8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

**Вывод:** яполучил практические навыки реализации классов на С++.