Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №1**

По дисциплине: “Алгоритмы и структуры данных”

Тема: «Классы и объекты в с++»**Вариант №10**

**Выполнил**:

студент 2 курса

группы ПО-7

Курмыса Е.Е.

**Проверила:**

Дряпко А. В.

Брест 2021

**Порядок выполнения работы**

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.

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

**Вариант задания:** структура – «Цех», поля: название цеха, начальник (оба – char\*), количество рабочих цеха (int).

**Ход работы**

workshop.h:

#ifndef WORKSHOP\_H

#define WORKSHOP\_H

const int NAME\_LENGTH = 50; // константа для строковых величин

class Workshop {

char name[NAME\_LENGTH];

char boss[NAME\_LENGTH];

int amount;

public:

Workshop(); // конструктор без аргументов

Workshop(const char\*, const char\*, int); // конструктор с аргументами

Workshop(const Workshop&); // конструктор копирования

~Workshop(); // деструктор

const char\* getName(); // получение того или иного поля

const char\* getBoss();

int getAmount();

void setName(const char\*); // замена того или иного поля на аргумент функции

void setBoss(const char\*);

void setAmount(int);

void set(const char\*, const char\*, int); // замена всех полей на аргументы функции

void show(); // вывод всех полей форматированным способом

void secret(); // тайная функция (для указателя на функцию)

};

#endif

workshop.cpp

#include <iostream>

#include "workshop.h"

using namespace std;

Workshop::Workshop() {

cout << "Executing a constructor without any parameters for object " << this << "...\n";

strcpy\_s(name, "Name");

strcpy\_s(boss, "Boss");

amount = 0;

cout << "A constructor without any parameters was executed for object " << this << "\n";

}

Workshop::Workshop(const char\* NAME, const char\* BOSS, int AMOUNT) {

cout << "Executing a constructor with parameters for object " << this << "...\n";

strcpy\_s(name, NAME);

strcpy\_s(boss, BOSS);

amount = AMOUNT;

cout << "A constructor with parameters was executed for object " << this << "\n";

}

Workshop::Workshop(const Workshop& workshop) {

cout << "Executing a copying constructor for object " << this << "...\n";

strcpy\_s(name, workshop.name);

strcpy\_s(boss, workshop.boss);

amount = workshop.amount;

cout << "A copying constructor was executed for object " << this << "\n";

}

Workshop::~Workshop() {

cout << "Executing a destructor for object " << this << "...\n";

cout << "A destructor was executed for object " << this << "\n";

}

const char\*

Workshop::getName() {

return name;

}

void

Workshop::setName(const char\* NAME) {

strcpy\_s(name, NAME);

}

const char\*

Workshop::getBoss() {

return boss;

}

void

Workshop::setBoss(const char\* BOSS) {

strcpy\_s(boss, BOSS);

}

int

Workshop::getAmount() {

return amount;

}

void

Workshop::setAmount(int AMOUNT) {

amount = AMOUNT;

}

void

Workshop::set(const char\* NAME, const char\* BOSS, int AMOUNT) {

strcpy\_s(name, NAME);

strcpy\_s(boss, BOSS);

amount = AMOUNT;

}

void

Workshop::show() {

cout << "Workshop's name is " << name << endl;

cout << "The boss is " << boss << endl;

cout << "The amount of workers is " << amount << endl;

}

void

Workshop::secret() {

strcpy\_s(name, "Absolute Happiness");

strcpy\_s(boss, "Nice");

amount = 69420;

}

Main.cpp

#include <iostream>

#include "workshop.h"

using namespace std;

int main() {

Workshop first; // конструктор без параметров

Workshop second("Workshop", "Zubenko Mikhail", 5928); // конструктор с параметрами

Workshop third = first; // конструктор копираования первого объекта

Workshop group[3] = { Workshop(), Workshop("Second", "Belyaev", 7), first }; // статический массив, внутри него все вышеперечисл. конструкторы

Workshop\* dynamic\_group; // динамический массив + указатель

dynamic\_group = new Workshop[2];

dynamic\_group[0].set("Steel", "Golubev", 19); // добавление в объект всего вместе

dynamic\_group[1].setAmount(5); // добавлени в объект по отдельности

dynamic\_group[1].setBoss("Dolubev");

dynamic\_group[1].setName("Piping");

void (Workshop:: \*secret)(); // указатель на функцию-компоненту

secret = &Workshop::secret;

cout << "The name of workshop No. 1 is " << first.getName() << endl;

first.setName("Concrete"); // изменение названия цеха

cout << "The name of workshop No. 1 is " << first.getName() << endl;

cout << "The boss of workshop No. 2 is " << second.getBoss() << endl;

second.setBoss("Kamyshov"); // изменение начальника

cout << "The boss of workshop No. 2 is " << second.getBoss() << endl;

cout << "The amount of workers in workshop No. 3 is " << third.getAmount() << endl;

third.setAmount(16); // изменение количества рабочих

cout << "The amount of workers in workshop No. 3 is " << third.getAmount() << endl;

group[0].show();

group[0].set("Name", "Boss", 100); // изменения всего через set

group[0].show();

dynamic\_group[1].show();

(dynamic\_group[1].\*secret)(); // тайная функция :)

dynamic\_group[1].show();

system("pause");

delete[] dynamic\_group; // очищаем динмаический массив dynamic\_group

}

**Вывод:** я получил практические навыки реализации классов в C++.