МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. И. С. ТУРГЕНЕВА»

Кафедра «Программная инженерия»

**ОТЧЁТ**

по лабораторной работе № 4

по дисциплине: «Объектно-ориентированное программирование

на С++»

Выполнил: Шорин В. Д. Шифр: 171406

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверил:

Отметка о зачете:

Дата: «\_\_» \_\_\_\_\_\_\_\_ 2018г.

Орёл. 2018

**Вариант 25**

**Задание**

Создайте базовый класс «Урок» и производные классы «Рисование», «Пение», «Биология».

Для производных классов разработать конструкторы с вызовом конструктора базового класса. Переопределить соответствующие методы (например, метод «Show» для отображения информации). Создать массив объектов базового класса и заполнить его объектами производных классов.

Разработать подпрограммы:

1. Отображение информации об уроках;
2. Подсчета количества объектов в каждой категории;
3. Отображения информации только из категории «Рисование», «Пение» или «Биология».

**Код:**

#include <iostream>#include <windows.h>#include <iomanip>#include <vector>

using namespace std;

class Lesson{

protected:

string type;

int typeLes;

public:

Lesson(string t, int tl){type = t; typeLes = tl;}

virtual void Show() { cout << "Тип: " << type << endl; }

string getType(){return type;}

int getTypeLes() {return typeLes;}

};

class Art : public Lesson{

string teacher; string smthAboutArt; string classType;

public:

Art (string CT, string T, int TL, string TEACHER, string SAA): Lesson(T, TL)

{ teacher = TEACHER; smthAboutArt = SAA; classType = CT; }

void Show() {

cout << "Предмет: " << classType << endl;

Lesson :: Show();

cout << "Преподаватель: "<< teacher << endl;

cout << "Какая-то информация: " << smthAboutArt << endl << endl;

}

string getTeacher(){return teacher;}

string getSmthAboutArt(){return smthAboutArt;}

};

class Singing : public Lesson{

string teacher; string song; string smthAboutSinging; string classType;

public:

Singing(string CT, string T, int TL, string TEACHER, string SONG, string SAS):Lesson(T, TL)

{ teacher = TEACHER; song = SONG; smthAboutSinging = SAS; }

void Show() {

cout << "Предмет: " << classType << endl;

Lesson :: Show();

cout << "Преподаватель: " << teacher << endl;

cout << "Песня: " << song << endl;

cout << "Какая-то информация: " << smthAboutSinging << endl << endl;

}

string getTeacher(){return teacher;} string getSong(){return song;}

string getSmthAboutSinging(){return smthAboutSinging;}

};

class Biology : public Lesson{

string teacher; string smthAboutBiology; string classType;

public:

Biology(string CT, string T, int TL, string TEACHER, string SAB):Lesson(T, TL)

{ teacher = TEACHER; smthAboutBiology = SAB; }

void Show() {

cout << "Предмет: " << classType << endl;

Lesson :: Show();

cout << "Преподаватель: "<< teacher << endl;

cout << "Какая-то информация: " << smthAboutBiology << endl << endl;

}

string getTeacher(){return teacher;}

string getSmthAboutBiology(){return smthAboutBiology;}

};

void cinGet(){ cin.get();}

int main(){

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int exit = 1; char userChoise; int artCount = 0; int singingCount = 0; int biologyCount = 0;

vector <Lesson\*> lesVect;

while (exit) {

system("cls");

cout << ">>Количество записей по рисованию: " << artCount << endl;

cout << ">>Количество записей по пению: " << singingCount << endl;

cout << ">>Количество записей по биологии: " << biologyCount << endl << endl;

cout << "Выберите действие: " << endl;

cout << ">>Добавление информации об уроке - 1" << endl;

cout << ">>Отображение информации об уроках - 2" << endl;

cout << ">>Отображение информации по категориям - 3" << endl << endl;

cout << ">>Выход - 0" << endl << endl;

cout << "Ваш выбор: "; cin >> userChoise;

switch(userChoise) {

case '1': {

system("cls"); char curUserChoise;

cout << "Выберите нужный урок:" << endl; cout << ">>Рисование - 1" << endl;

cout << ">>Пение - 2" << endl; cout << ">>Биология - 3" << endl << endl;

cout << ">>Назад - 9" << endl << endl; cout << "Ваш выбор: ";

cin >> curUserChoise; switch(curUserChoise) {

case'1': {

system("cls"); string curType; string curTeacher;

string curInfo; cout << "Введите информацию: " << endl;

cout << "Тип: "; cin.get(); getline(cin, curType);

cout << endl << "Преподаватель: "; getline(cin, curTeacher);

cout << endl << "Различная информация: "; getline(cin, curInfo);

lesVect.push\_back(new Art("Рисование", curType, 1, curTeacher, curInfo));

artCount++;

system("pause");

break;

}

case'2': {

system("cls");

string curType; string curTeacher; string curSong;

string curInfo; cout << "Введите информацию: " << endl;

cout << "Тип: "; cin.get(); getline(cin, curType);

cout << endl << "Преподаватель: "; getline(cin, curTeacher);

cout << endl << "Песня: "; getline(cin, curSong);

cout << endl << "Различная информация: "; getline(cin, curInfo);

lesVect.push\_back(new Singing("Пение", curType, 2, curTeacher, curSong, curInfo));

singingCount++; system("pause");

break; }

case '3': {

system("cls"); string curType; string curTeacher;

string curInfo;

cout << "Введите информацию: " << endl;

cout << "Тип: "; cin.get(); getline(cin, curType);

cout << endl << "Преподаватель: "; getline(cin, curTeacher);

cout << endl << "Различная информация: "; getline(cin, curInfo);

lesVect.push\_back(new Biology("Биология", curType, 3, curTeacher, curInfo));

biologyCount++; system("pause");

break;

}

case '9':break; } break; }

case '2': {

system("cls");

for (int i = 0; i < lesVect.size(); i++) {

lesVect[i]->Show();

}

system("pause");

break;

}

case '3': {

system("cls"); char curUserChoise; cout << "Выберите нужный урок:" << endl;

cout << ">>Рисование - 1" << endl; cout << ">>Пение - 2" << endl;

cout << ">>Биология - 3" << endl << endl; cout << ">>Назад - 9" << endl << endl;

cout << "Ваш выбор: "; cin >> curUserChoise;

switch(curUserChoise)

{

case '1': {

for(int i = 0; i < lesVect.size(); i++) {

if (lesVect[i]->getTypeLes() == 1) {

lesVect[i]->Show();

}

}

break;

}

case '2': {

for(int i = 0; i < lesVect.size(); i++) {

if (lesVect[i]->getTypeLes() == 2) {

lesVect[i]->Show();

}

}

break;

}

case '3': {

for(int i = 0; i < lesVect.size(); i++) {

if (lesVect[i]->getTypeLes() == 3) {

lesVect[i]->Show();

}

}

break;

}

case '9': { break; }

}

system("pause");

break;

}

case'0': { exit = 0; break; }

default:

cout << "Нет такой команды" << endl;

system("pause");

break;

}

}

lesVect.clear();

return 0;

}