# ЛАБОРАТОРНАЯ РАБОТА № 25

# РАЗРАБОТКА И ОТЛАДКА АЛГОРИТМОВ И ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ МНОЖЕСТВЕННОГО НАСЛЕДОВАНИЯ

Цель работы:

Применить на практике знания:

* по созданию производных классов, образованных от нескольких базовых;
* по вызову методов базовых классов.

## КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Наследование является одной из главных особенностей ООП. Механизм наследования классов позволяет строить иерархии, в которых производные классы получают элементы родительских, или базовых, классов и могут дополнять их или изменять их свойства.

### Множественное наследование

Множественное наследование означает, что класс имеет несколько базовых классов. Если в базовых классах есть одноименные элементы, при этом может произойти конфликт идентификаторов, который устраняется с помощью операции доступа к области видимости.

Если базовых классов несколько, они перечисляются через запятую. Ключ доступа может стоять перед каждым классом, например:

class A {...}; class В {...}; class С {...};

class D: A, protected В, public С {...};

Модификаторы доступа при наследовании:



зависимости от ключа. Обращение к ним может осуществляться только через методы базового класса.

Элементы protected при наследовании с ключом private становятся в производном классе private, в остальных случаях права доступа к ним не изменяются.

Доступ к элементам public при наследовании становится соответствующим ключу доступа.

Если базовый класс наследуется с ключом private, можно выборочно сделать некоторые его элементы доступными в производном классе, объявив их в секции public производного класса с помощью операции доступа к области видимости:

class Base{ …

public: void f();

};

class Derived : private Base{

…

public: Base::void f();

};

## ПРИМЕРЫ

**Множественное наследование**

Написать программу, в которой классы Manager и Scientist являются производными от двух базовых: классов Worker и Student.

*UML-*

*диаграмма*

*классов*

*для*

**Worker**

**Student**

**fio**

**school**

**code**

**de**

**gree**

**salary**

getdata()

getedu()

putdata()

putedu()

**Manager**

**category**

getdata()

putdata()

**Scientist**

**pubs**

getdata()

putdata()

#include <iostream>

#include <Windows.h>

using namespace std;

class Worker {

char fio[30];

int code;

double salary;

public:

void getdata();

void putdata() const {

cout << "\n Фамилия: " << fio << "\n Номер: " << code << "\n Зарплата: " << salary;

}

};

void Worker::getdata() {

cout << "\n Введите фамилию: ";

cin >> fio;

cout << " Введите номер: ";

cin >> code;

cout << " Введите зарплату: ";

cin >> salary;

}

//////////////////////////////////////////////////////////////

class Student {

char school[20];

char degree[20];

public:

void getedu();

void putedu() const {

cout << "Учебное заведение : " << school;

cout << "\nСтепень образования : " << degree;

};

};

void Student::getedu() {

cout << "Введите учебное заведение : ";

cin >> school;

cout << "Введите степень образования : ";

cin >> degree;

}

//////////////////////////////////////////////////////////////

class Manager : private Worker, private Student {

char category[20];

public:

void getdata() {

Worker::getdata();

Student::getedu();

cout << "Категория: ";

cin >> category;

}

void putdata() const {

Worker::putdata();

Student::putedu();

cout << "Категория: " << category;

}

};

//////////////////////////////////////////////////////////////

class Scientist : private Worker, private Student {

int pubs;

public:

void getdata() {

Worker::getdata();

Student::getedu();

cout << "Количество публикаций : ";

cin >> pubs;

}

void putdata() const {

Worker::putdata();

Student::putedu();

cout << "Количество публикаций : " << pubs;

}

};

//////////////////////////////////////////////////////////////

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Manager m1;

Scientist s1;

m1.getdata();

s1.getdata();

cout << "Информация о сотрудниках : \n";

m1.putdata();

cout << endl;

s1.putdata();

system("pause");

return 0;

}

## ЗАДАНИЯ

Для выполнения задания внесите дополнения в Лабораторную работу №24. Варианты заданий остаются прежними. ***Вариант 1***

Задание: Множественное наследование.

Разработайте базовый класс CLIENT. Элемент класса:

* ФИО;
* количество поездок.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс SKIDKA – производный от класса REIS и CLIENT. В этот класс добавьте метод, в котором проверяется количество поездок. Если количество> 10, то выводится на экран сообщение «Скидка 5%» и цена билета с учетом скидки, в противном случае – 100% стоимость.

### Вариант 2

Задание: Множественное наследование.

Разработайте базовый класс ZNAK. Элемент класса:

* знак зодиака.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс FREND – производный от класса NOTE1 и ZNAK. В этот класс добавьте метод, в котором проверяется введенная дата. Если дата совпала со значением поля дата рождения, то выводится на экран сообщение «Отправить поздравление» и электронный адрес.

### Вариант 3

Задание: Множественное наследование. Разработайте базовый класс CLIENT.

Элемент класса:

* ФИО;
* количество поездок.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс SKIDKA – производный от класса REIS и CLIENT. В этот класс добавьте метод, в котором проверяется количество поездок. Если количество> 10, то выводится на экран сообщение «Скидка 5%» и цена билета с учетом скидки, в противном случае – 100% стоимость.

### Вариант 4

Задание: Множественное наследование.

Разработайте базовый класс SALARY. Элемент класса:

* оклад.

Методы:

метод, осуществляющий ввод значений полей класса с клавиатуры;

метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс SOTRUDNIK – производный от класса WORKER1 и SALARY. В этот класс добавьте метод, в котором проверяется нагрузка и количество отработанных часов. Если значения полей совпали, то выводится на экран 100% зарплаты, в противном случай – 95% зарплаты.

### Вариант 5

Задание: Множественное наследование.

Разработайте базовый класс WEEK. Элемент класса:

* день недели.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс VIZIT – производный от класса NOTE1 и WEEK. В этот класс добавьте метод, в котором проверяется значение дня недели. Если день недели приходится на воскресенье, то выводится на экран сообщение «Перенести не другой день» и информация о встрече.

### Вариант 6

Задание: Множественное наследование.

Разработайте базовый класс SKIDKA. Элемент класса:

* размер скидки.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс AKCIA – производный от класса PRICE1 и SKIDKA. В этот класс добавьте метод, в котором проверяется остаток товара. Если остаток <5 единиц, то выводится на экран стоимость с учетом скидки, в противном случае – 100% стоимость.

### Вариант 7

Задание: Множественное наследование.

Разработайте базовый класс SHTRAF.

Элементы класса:

* количество штрафов;
* общая сумма штрафов.

Методы:

* метод, осуществляющий ввод
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс VLADELEC – производный от класса AVTO1 и SHTRAF. В этот класс добавьте метод, который выводит на экран всю информацию о владельце и автомобиле.

### Вариант 8

Задание: Множественное наследование.

Разработайте базовый класс YEAR. Элемент класса:

* знак года.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс FREND – производный от класса ZNAK1 и YEAR. В этот класс добавьте метод, в котором проверяется введенная дата. Если дата совпала со значением поля дата рождения, то выводится на экран сообщение «Позвонить и поздравить» и телефон.

### Вариант 9

Задание: Множественное наследование.

Разработайте базовый класс СLIENT. Элемент класса:

* номер паспорта;
* количество поездок.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс AKCIA – производный от класса BILET и СLIENT. В этот класс добавьте метод, который при каждой десятой поездке выводит на экран сообщение «Бесплатная поездка» и информацию о пассажире.

### Вариант 10

Задание: Множественное наследование.

Разработайте базовый класс HOTEL. Элемент класса:

* количество номеров;
* количество свободных номеров.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс AKCIA – производный от класса СLIENT и HOTEL. В этот класс добавьте метод, который при каждой двадцатой остановке в отеле выводит на экран сообщение «Бесплатное обслуживание» и информацию о клиенте.

### Вариант 11

Задание: Множественное наследование.

Разработайте базовый класс POSTAVKA. Элемент класса:

* название фирмы-поставщика;
* цена поставщика.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс NAVAR – производный от класса TOVAR1 и POSTAVKA. В этот класс добавьте метод, который рассчитывает прибыль и выводит на экран информацию о товаре.

### Вариант 12

Задание: Множественное наследование.

Разработайте базовый класс READER. Элемент класса:

* ФИО.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс INFO – производный от класса DATA и READER. В этот класс добавьте метод, в котором проверяется дата. Если книга взята более 20 дней назад, то выводится на экран сообщение «Позвонить и напомнить» и информация о читателе.

### Вариант 13

Задание: Множественное наследование.

Разработайте базовый класс FACULTET. Элемент класса:

* название.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс REZULT – производный от класса ABITURIENT1 и FACULTET. В этот класс добавьте метод, в котором проверяется средний балл. Если средний балл <3, то выводится на экран сообщение «Вы не поступили», если>=3 – выводится информация о человеке.

### Вариант 14

Задание: Множественное наследование.

Разработайте базовый класс BANK. Элемент класса:

* название банка получателя.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс RENTA – производный от класса ORDER1 и BANK. В этот класс добавьте метод, в котором сравниваются названия банка отправителя и банка получателя. Если они совпадают, то выводится на экран сообщение «Без комиссии», если нет – выводится сообщение «Комиссия 0.5%» и оставшаяся сумма.

### Вариант 15

Задание: Множественное наследование.

Разработайте базовый класс VUZ. Элемент класса:

* название ВУЗа;
* количество мест.

Методы:

* метод, осуществляющий ввод значений полей класса с клавиатуры;
* метод, осуществляющий вывод значений полей класса на экран.

Разработайте класс NABOR – производный от класса SCHOOL и VUZ. В этот класс добавьте метод, в котором сравниваются количество выпускников и количество мест в ВУЗе. Если количество выпускников меньше, то выводится на экран сообщение «Недобор», если нет – выводится сообщение «Конкурс» и количество человек на место.

## КОНТРОЛНЫЕ ВОПРОСЫ

1. Поясните, чем отличаются различные типы наследования: public, protected, private.
2. Объясните, что представляет собой множественное наследование.
3. Приведите синтаксис описания производного класса при множественном наследовании.