# **3. Лабораторная работа № 3. Изучение возможностей наследования классов**

## **3.1. Цель и задачи работы, требования к результатам ее выполнения**

Цель работы состоит в овладении навыками разработки программ на языке Си++, использующих возможности наследования классов для решения различных задач. Для достижения цели необходимо выполнить следующие задачи:

- изучить необходимые учебные материалы, посвященные наследованию классов в языке Си++;

- разработать программу на языке Си++ для решения заданного варианта задания;

- отладить программы;

- представить результаты работы программы;

- подготовить отчет по лабораторной работе.

## **3.2. Краткая характеристика объекта изучения**

### **3.2.1 Общие сведения о наследовании классов**

Основная идея, используемая при наследовании классов, заключается в том, что на основе существующего класса (класс родитель или базовый класс), создается производный класс (класс- наследник, дочерний класс), который включает в себя поля и функции базового класса (наследует их), и содержит дополнительные поля (обладает новыми свойствами) и функции.

Примеры определения производных классов:

*class S: X, Y, Z*

*{ ... } ;*

*class B: public A*

*{….};*

*class D: public X, protected B*

*{….};*

### **3.2.2 Статусы доступа при наследовании классов**

Перед именем базового класса может указываться статус доступа наследования (одно из ключевых слов *public, protected, private*). Статус доступа наследования определяет статус доступа наследуемых полей и функций из базового класса внутри производного класса.

Производный класс может определяться с ключевым словом *struct* или *class* (с ключевым словом *union* производный класс не определяется). Если производный класс определен с ключевым словом *class*, то по умолчанию статус доступа наследования *private*.

Если производный класс определен с ключевым словом *struct*, то по умолчанию статус доступа наследования *public*.

Статусы доступа при наследовании классов представлены в таблице 1.

Таблица 1- Статусы доступа при наследовании классов



### **3.2.3. Особенности конструкторов при наследовании**

Конструктор производного класса в первую очередь всегда должен вызывать конструктор базового класса. Если это действие не выполняется явно, то по умолчанию вызывается конструктор без параметров (если он есть, если его нет, будет ошибка). Если класс имеет несколько базовых, то конструкторы базовых классов должны вызываться в порядке перечисления этих классов в списке базовых.

### **3.2.4. Особенности деструкторов при наследовании**

Деструктор производного класса всегда неявно по умолчанию после выполнения своего тела вызывает деструкторы базовых классов. Причем порядок разрушения объекта (вызовов деструкторов) обратен порядку создания (вызова конструкторов).

### **3.2.5. Переопределение функций. Виртуальные функции**

Если в производном классе объявлена функция с именем, типом возвращаемого значения и количеством и типами параметров, такая же, как в базовом классе, то данная функция является переопределенной. (Не путать с перегрузкой функций). С помощью простых переопределенных функций реализуется механизм статического полиморфизма. (Полиморфизм – возможность функции в производном классе работать по-другому).

Суть статического связывания: когда указатель одного типа ссылается на объект другого типа при наследовании классов, то выбор переопределенного метода определяется типом указателя, а не типом объекта.

Суть динамического связывания: когда указатель одного типа ссылается на объект другого типа при наследовании классов, то выбор переопределенного метода определяется типом объекта, а не типом указателя, для этого переопределенный метод должен быть объявлен виртуальным в базовом классе. С помощью динамического связывания реализуется механизм динамического полиморфизма.

Динамический полиморфизм при переопределении метода в производном классе обеспечивается объявлением заголовка метода с ключевым словом virtual в базовом классе. Встретив у функции модификатор *virtual*, компилятор создает для класса таблицу виртуальных функций, а в класс добавляет новый скрытый для программиста член — указатель на эту таблицу.

Таблица виртуальных функций хранит в себе адреса всех виртуальных методов класса (по сути, это массив указателей), а также всех виртуальных методов базовых классов этого класса. За счет этих указателей на функции обеспечивается динамическое связывание: при вызове метода через указатель, который имеет тип базового класса, но содержит адрес объекта производного класса (такое преобразование допустимо и выполняется неявно), вызывается именно метод производного класса (вызов метода определяется типом объекта).

## **3.3. Задачи и порядок выполнения работы**

В работе первоначально необходимо создать базовый класс и на его основе создать производный класс. При этом необходимо использовать вызов конструктора базового класса внутри конструктора производного класса, а также возможности переопределения методов (функций) класса. Необходимо разобраться с понятием виртуального метода, а также статическим и динамическим полиморфизмом. Продемонстрировать данные возможности в выполняемом примере.

## **3.4. Форма отчета по лабораторной работе**

Отчет должен содержать: титульный лист, цель работы, условие задачи, текст программы с комментариями, вывод результатов работы программы, выводы по работе.

## **3.5. Вопросы для самоконтроля**

1. Производные классы, статусы доступа наследуемых полей и методов в производных классах.

2. Особенности конструкторов с производных классах.

3. Особенности деструкторов с производных классах.

4. Переопределение функций в производных классах, виртуальные функции.

5. Статическое и динамическое связывание.

6. Ключевое слово final.

7. Ключевое слово override.

8. Множественное наследование и виртуальные базовые классы.

9. Абстрактные и локальные классы.

**Варианты заданий для лабораторной работы № 3**

Вариант № 1

Создать базовый класс «вектор на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*), определяющей конец вектора (начало вектора находится в точке с координатами 0, 0); конструктор для инициализации полей; функция для вычисления длины вектора, функция для печати полей и длины вектора. Создать производный класс «вектор в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для вычисления длины вектора; переопределенная функция для печати полей и длины вектора. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 2

Создать базовый класс «точка на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*); конструктор для инициализации полей; функция для печати значений полей. Создать производный класс «точка в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для печати значений полей (внутри переопределенной функции в первую очередь должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 3

Создать базовый класс «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления площади квадрата; функция для печати поля и площади квадрата. Создать производный класс «куб». Элементы класса: конструктор для инициализации поля; переопределенная функция для вычисления объема куба (вместо площади) (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 4

Создать базовый класс «прямоугольник». Элементы класса: поля, задающие длины сторон (статус доступа *protected*); конструктор для инициализации полей; функция для вычисления площади прямоугольника; функция для печати полей и значения площади. Создать производный класс «прямоугольный параллелепипед». Элементы класса: дополнительное поле, задающее высоту; конструктор для инициализации полей; переопределенная функция для вычисления объема (вместо площади) (внутри переопределенной функции должна вызываться функция из базового класса); переопределенная функция для печати полей и значения объема. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 5

Создать базовый класс «круг». Элементы класса: поле, задающее радиус; конструктор для инициализации поля (статус доступа *protected*); функция для вычисления площади круга (площадь круга ); функция для печати полей и площади. Создать производный класс «шар». Элементы класса: конструктор для инициализации поля; переопределенная функция для вычисления объема (вместо площади круга) шара (площадь шара ). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 6

Создать базовый класс «автомобиль». Элементы класса: поле, содержащее наименование модели автомобиля; поле, содержащее значение максимальной скорости (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров автомобиля. Создать производный класс «грузовой автомобиль». Элементы класса: дополнительно поле, содержащее грузоподъемность автомобиля в тоннах; конструктор для инициализации полей; переопределенная функция печати параметров автомобиля (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 7

Создать базовый класс «вещественное число». Элементы класса: поле, задающее значение числа (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления модуля числа; функция для печати поля и модуля числа. Создать производный класс «комплексное число». Элементы класса: дополнительно поле, задающее значение мнимой части числа; конструктор для инициализации полей; переопределенная функция для вычисления модуля числа (модуль числа – корень квадратный из суммы квадратов вещественной и мнимой частей числа); переопределенная функция для печати полей и модуля числа. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 8

Создать базовый класс «вектор на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*), определяющей конец вектора (начало вектора находится в точке с координатами 0, 0); конструктор для инициализации полей; функция для печати координат вектора. Создать производный класс «вектор в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для печати координат вектора (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 9

Создать базовый класс «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления периметра квадрата; функция для печати длины стороны и периметра. Создать производный класс «прямоугольник». Элементы класса: дополнительное поле, задающее другую сторону; конструктор для инициализации полей; переопределенная функция для вычисления периметра прямоугольника; переопределенная функция для печати длин сторон и периметра. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 10

Создать базовый класс «автомобиль». Элементы класса: поле, содержащее наименование модели автомобиля; поле, содержащее значение максимальной скорости (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров автомобиля. Создать производный класс «автобус». Элементы класса: дополнительно поле, содержащее максимальное число перевозимых пассажиров; конструктор для инициализации полей; переопределенная функция печати параметров автобуса (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 11

Создать базовый класс «школа». Элементы класса: поле, содержащее название школы; поле, содержащее значение числа обучаемых в школе (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров школы. Создать производный класс «специализированная школа». Элементы класса: дополнительно поле, содержащее название специализации школы; конструктор для инициализации полей; переопределенная функция печати параметров школы (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 12

Создать базовый класс «круг». Элементы класса: поле, содержащее значение радиуса круга (статус доступа *protected*); конструктор для инициализации поля; функция для печати радиуса круга. Создать производный класс «эллипс». Элементы класса: дополнительно поле, содержащее значение второй полуоси эллипса (для задания первой полуоси использовать наследуемое поле радиуса круга); конструктор для инициализации полей; переопределенная функция печати параметров эллипса (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 13

Создать базовый класс - сотрудник предприятия. Компоненты класса: поля: ФИО, оклад, надбавка за стаж (в процентах от оклада за 1 год), стаж (в годах), статус доступа полей *protected*;

конструктор для инициализации полей;

функция для вычисления зарплаты;

функция для печати параметров сотрудника.

Создать производный класс - начальник подразделения.

Дополнительные поля: процентная надбавка к окладу за выполнение обязанностей начальника и название подразделения.

Переопределить функцию для вычисления зарплаты и функцию для печати параметров начальника. Внутри переопределенных функций вызывать соответствующие функции из базового класса.

Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 14

Создать базовый класс – простой счет в банке. Компоненты класса: поля: ФИО владельца, начальная сумма счета, ставка вклада (проценты в год), время существования вклада в годах, статус доступа полей *protected*;

конструктор для инициализации полей;

функция для вычисления суммы на счете с учетом начисленных процентов за время существования вклада;

функция для печати параметров счета.

Создать производный класс – привилегированный счет.

Дополнительные поля: процент кредита предоставляемому по счету (проценты от доступной на счете суммы с учетом времени существования вклада).

Переопределенная функция для вычисления суммы на счете с учетом доступного кредита.

Переопределенная функция для печати параметров счета.

Внутри переопределенных функций вызывать соответствующие функции из базового класса.

Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 15

Создать базовый класс «вектор на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*), определяющей конец вектора (начало вектора находится в точке с координатами 0, 0); конструктор для инициализации полей; функция для вычисления длины вектора, функция для печати полей и длины вектора. Создать производный класс «вектор в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для вычисления длины вектора; переопределенная функция для печати полей и длины вектора. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 16

Создать базовый класс «точка на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*); конструктор для инициализации полей; функция для печати значений полей. Создать производный класс «точка в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для печати значений полей (внутри переопределенной функции в первую очередь должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 17

Создать базовый класс «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления площади квадрата; функция для печати поля и площади квадрата. Создать производный класс «куб». Элементы класса: конструктор для инициализации поля; переопределенная функция для вычисления объема куба (вместо площади) (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 18

Создать базовый класс «прямоугольник». Элементы класса: поля, задающие длины сторон (статус доступа *protected*); конструктор для инициализации полей; функция для вычисления площади прямоугольника; функция для печати полей и значения площади. Создать производный класс «прямоугольный параллелепипед». Элементы класса: дополнительное поле, задающее высоту; конструктор для инициализации полей; переопределенная функция для вычисления объема (вместо площади) (внутри переопределенной функции должна вызываться функция из базового класса); переопределенная функция для печати полей и значения объема. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 19

Создать базовый класс «круг». Элементы класса: поле, задающее радиус; конструктор для инициализации поля (статус доступа *protected*); функция для вычисления площади круга (площадь круга ); функция для печати полей и площади. Создать производный класс «шар». Элементы класса: конструктор для инициализации поля; переопределенная функция для вычисления объема (вместо площади круга) шара (площадь шара ). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 20

Создать базовый класс «автомобиль». Элементы класса: поле, содержащее наименование модели автомобиля; поле, содержащее значение максимальной скорости (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров автомобиля. Создать производный класс «грузовой автомобиль». Элементы класса: дополнительно поле, содержащее грузоподъемность автомобиля в тоннах; конструктор для инициализации полей; переопределенная функция печати параметров автомобиля (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 21

Создать базовый класс «вещественное число». Элементы класса: поле, задающее значение числа (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления модуля числа; функция для печати поля и модуля числа. Создать производный класс «комплексное число». Элементы класса: дополнительно поле, задающее значение мнимой части числа; конструктор для инициализации полей; переопределенная функция для вычисления модуля числа (модуль числа – корень квадратный из суммы квадратов вещественной и мнимой частей числа); переопределенная функция для печати полей и модуля числа. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 22

Создать базовый класс «вектор на плоскости». Элементы класса: поля, задающие координаты точки (статус доступа *protected*), определяющей конец вектора (начало вектора находится в точке с координатами 0, 0); конструктор для инициализации полей; функция для печати координат вектора. Создать производный класс «вектор в трехмерном пространстве». Элементы класса: дополнительное поле, задающее дополнительную координату; конструктор для инициализации полей; переопределенная функция для печати координат вектора (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 23

Создать базовый класс «квадрат». Элементы класса: поле, задающее длину стороны (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления периметра квадрата; функция для печати длины стороны и периметра. Создать производный класс «прямоугольник». Элементы класса: дополнительное поле, задающее другую сторону; конструктор для инициализации полей; переопределенная функция для вычисления периметра прямоугольника; переопределенная функция для печати длин сторон и периметра. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 24

Создать базовый класс «автомобиль». Элементы класса: поле, содержащее наименование модели автомобиля; поле, содержащее значение максимальной скорости (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров автомобиля. Создать производный класс «автобус». Элементы класса: дополнительно поле, содержащее максимальное число перевозимых пассажиров; конструктор для инициализации полей; переопределенная функция печати параметров автобуса (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 25

Создать базовый класс «школа». Элементы класса: поле, содержащее название школы; поле, содержащее значение числа обучаемых в школе (статус доступа *protected*); конструктор для инициализации полей; функция для печати параметров школы. Создать производный класс «специализированная школа». Элементы класса: дополнительно поле, содержащее название специализации школы; конструктор для инициализации полей; переопределенная функция печати параметров школы (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 26

Создать базовый класс «круг». Элементы класса: поле, содержащее значение радиуса круга (статус доступа *protected*); конструктор для инициализации поля; функция для печати радиуса круга. Создать производный класс «эллипс». Элементы класса: дополнительно поле, содержащее значение второй полуоси эллипса (для задания первой полуоси использовать наследуемое поле радиуса круга); конструктор для инициализации полей; переопределенная функция печати параметров эллипса (внутри переопределенной функции должна вызываться функция из базового класса). Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 27

Создать базовый класс - сотрудник предприятия. Компоненты класса: поля: ФИО, оклад, надбавка за стаж (в процентах от оклада за 1 год), стаж (в годах), статус доступа полей *protected*;

конструктор для инициализации полей;

функция для вычисления зарплаты;

функция для печати параметров сотрудника.

Создать производный класс - начальник подразделения.

Дополнительные поля: процентная надбавка к окладу за выполнение обязанностей начальника и название подразделения.

Переопределить функцию для вычисления зарплаты и функцию для печати параметров начальника. Внутри переопределенных функций вызывать соответствующие функции из базового класса.

Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.

Вариант № 28

Создать базовый класс – простой счет в банке. Компоненты класса: поля: ФИО владельца, начальная сумма счета, ставка вклада (проценты в год), время существования вклада в годах, статус доступа полей *protected*;

конструктор для инициализации полей;

функция для вычисления суммы на счете с учетом начисленных процентов за время существования вклада;

функция для печати параметров счета.

Создать производный класс – привилегированный счет.

Дополнительные поля: процент кредита предоставляемому по счету (проценты от доступной на счете суммы с учетом времени существования вклада).

Переопределенная функция для вычисления суммы на счете с учетом доступного кредита.

Переопределенная функция для печати параметров счета.

Внутри переопределенных функций вызывать соответствующие функции из базового класса.

Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.