# **1. Лабораторная работа № 1. Изучение классов языка Си++**

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

Цель работы состоит в изучении основных понятий объектно- ориентированного программирования языка Си++ – классов и объектов, и овладении навыками разработки программ на языке Си++ с использованием объектно- ориентированных средств. Для достижения цели необходимо выполнить следующие задачи:

- изучить необходимые учебные материалы, посвященные основам объектно- ориентированного программирования на языке Си++;

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

- отладить программу;

- выполнить решение контрольного примера с помощью программы и ручной расчет контрольного примера;

- подготовить отчет по лабораторной работе.

## **1.2. Краткая характеристика объекта изучения**

### **1.2.1. Понятие класса и объекта**

Класс в языке Си++ – это новый тип, определяемый программистом, включающий данные (поля класса) и методы (функции) для обработки этих данных. Переменные этого типа называются объектами.

Формат объявления класса:

*<Ключевое слово> <Имя\_класса>*

*{*

*<список компонент>*

*};*

В качестве ключевого слова используется одно из трех ключевых слов:

*- struct*

*- class*

*- union*

Можно дать такое определение класса через структуру, которая была в языке Си. Класс – это структура, в которую введены методы для обработки полей.

Объекты – это переменные типа класса. Формат определения объектов:

*<Имя\_класса> <Имя\_объекта1>,…<Имя\_объекта\_N>;*

Обращение к полям и методам класса внутри методов класса просто по имени, а за пределами класса через имя объекта и операцию «.» или через имя указателя на объект и операцию «->». Каждый объект класса имеет в оперативной памяти свои копии полей класса.

Пример работы с полями:

*struct A { int i; void print() { printf(“i=%d”, i); } };*

*A a1; A \*pA=&a1;*

*a1.i=10; a1.print(); pA->i=10; pA->print();*

*a1.A::i=10; a1.A::print(); pA->A::i=10; pA->A::print();*

### **1.2.2. Доступность компонент класса**

Свойство доступности определяет возможность доступа к полям и методам за пределами класса (через имя объекта или через указатель на объект).

Существуют три статуса доступа:

* *public* (полностью доступны за пределами класса);
* *private* (не доступны за пределами класса, можно обращаться к компонентам только в методах своего класса);
* *protected* (доступны только в своем классе и в производных классах).

По умолчанию, если класс определен с ключевым словом *struct*, то все компоненты имеют статус доступа *public*. Если с ключевым словом *union*, тоже *public*, но все поля каждого объекта располагаются в памяти, начиная с одного адреса. Если класс определен с ключевым словом *class*, то все поля и методы по умолчанию имеют статус доступа *private*.

Статус доступа можно изменить с помощью соответствующих модификаторов, что продемонстрировано в следующем примере:

*struct A*

*{*

*…… // Статус доступа public*

*private:*

*………. // Статус доступа private*

*protected:*

*……….// Статус доступа protected*

*};*

*class B*

*{*

*…… // Статус доступа private*

*public:*

*………. // Статус доступа public*

*protected:*

*……….// Статус доступа protected*

*};*

### **1.2.3. Основные элементы класса**

***Компонентные данные и функции класса***

Компонентные данные (поля класса) и функции класса (методы класса) уже во многом рассмотрены выше. Следует добавить, что методы класса могут быть определены внутри класса, в этом случае они по возможности (если нет ограничений) являются подставляемыми, но чаще всего класс содержит описание (заголовки) методов, а определения методов находятся за пределами класса. Эта возможность является удобной в проектах, состоящих из многих файлов. Создается отдельный файл с расширением .h (заголовочный файл), в котором находится описание класса вместе с полями и заголовками методов. Определения методов находится в файле реализации класса (файле с расширением .cpp). В этом случае, чтобы использовать класс в другом файле с исходным кодом, необходимо подключить заголовочный файл с описанием класса. Также внутри методов можно использовать умалчиваемые значения параметров (требования такие же, как к обычным функциям). В классах возможна перегрузка методов.

В следующем примере показаны перечисленные возможности:

*struct Complex*

*{*

*double real, image; // Поля класса*

*void define(double re=0.0, double im=0.0) // Определение метода*

*// внутри класса*

*{*

*real=re; image=im;// Обращение к полям внутри метода*

*}*

*void print(); // Описание метода*

*};*

*void Complex::print() // Определение метода за пределами класса*

*{*

*printf("\nreal=%f image=%f", real, image);*

*}*

***Конструктор класса***

Конструктор класса – специальный блок операторов (инструкций), вызываемый при создании объекта. Назначение: присвоение начальных значений полям, выделение памяти, открытие файлов, сетевых соединений и т.п. Имя конструктора совпадает с именем класса, конструктор не имеет возвращаемого значения. Возможна перегрузка конструкторов. Конструктор может определяться как внутри класса, так и за пределами.

Формат определения конструктора внутри класса:

*Имя\_класса(Список\_формальных\_параметров)*

*{ Операторы\_тела конструктора }*

По умолчанию класс всегда имеет конструктор копирования вида *A(A& a) {… }* (*A* – имя класса), создающий копию объекта (происходит копирование полей), и если нет явного конструктора, то по умолчанию создается конструктор без параметров. Эти конструкторы можно переопределять.

Примеры вызовов конструкторов:

*A a1; A\* pA=new A; // Вызываются конструкторы без параметров*

*A a2(3, 4); A \* pA2=new A(3, 4);*

// Вызываются конструкторы с 2-мя параметрами

Для конструктора с одним параметром можно использовать форму:

*A a1=5; A a2=a1;* вместо *A a1(5); A a2(a1);*

***Деструктор класса***

Деструктор – специальный блок операторов (инструкций), служащий для деинициализации объекта (освобождение памяти, закрытие файлов и т.п.).

Вызывается автоматически при удаление объекта, например, оператором *delete* или при выходе из блока, в котором существует объект. Не имеет возвращаемого значения и параметров. Может определяться как внутри класса, так и за пределами. Пример деструктора:

*~имя\_класса()*

*{*

*тело\_деструктора*

*}*

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

В работе необходимо разобраться с понятием класса некоторой предметной области и соответствующему ему класса как типа языка Си++, введенного пользователем. Также необходимо знать отличие классов от объектов и назначение и порядок использования основных элементов класса в Си++: полей, методов (функций класса), конструкторов, деструктора. Обратить внимание на способы создания массива объектов класса динамически.

Студент разрабатывает программу на языке Си++ в виде консольного приложения, ***в программе необходимо создать массив и последовательной контейнер STL*** (vector, deque или list) объектов некоторого класса, данные об объектах читаются из текстового файла. Результаты работы программы и результаты ручного расчета (при необходимости) представляет преподавателю в отчете.

***Входные данные*** должны читаться из файла.

Также ***разрабатываемое ПО должно состоять*** из трёх файлов – файла с объявлением класса (.hpp файл), файла с определением класса (.cpp файл) и файла с функцией main.

Во всех вариантах также необходимо:

Описать класс, включающий заданные поля и методы (функции). Разработать две программы, одна из которых создает массив объектов, а другая последовательный контейнер STL (vector, deque или list выбрать самостоятельно) объектов и выполняет требуемые действия. **Все исходные данные для работы программы читаются из текстового файла, созданного в простом редакторе типа «Блокнот».** Выходные данные выводятся **на консоль и в текстовый файл, для этого использовать одну функцию для вывода**.

## **1.4 Важное пояснение**

Следует отметить, что для создания массива объектов динамически, когда существует конструктор с параметрами для инициализации полей объекта, можно использовать два основных способа.

Особенности первого способа:

- необходимо в класс включить конструктор без параметров и отдельный метод для инициализации полей, например, с именем *set* и числом параметров, которые соответствуют числу инициализируемых полей класса;

- объявляем указатель на массив объектов, если имя класса *MyClass*, то это объявление может иметь вид: *MyClass \*pOb;*

- задаем число объектов, например, вводим с клавиатуры или читаем из файла как значение целой переменой *n*, и создаем массив объектов динамически *pOb=new MyClass[n];* здесь, при создании каждого объекта, вызывается конструктор без параметров, именно для этого его и нужно включить в класс;

- в цикле для каждого объекта вводим параметры или читаем из файла для его инициализации и инициализируем уже созданный объект с помощью вызова метода *set*, это действие может иметь вид *pOb[i].set(…);* где в метод *set* вместо многоточия передаются фактические параметры.

Особенности второго способа (здесь не нужен конструктор без параметров и метод *set*):

- объявляем указатель на указатель на массив объектов, если имя класса *MyClass*, то это объявление может иметь вид: *MyClass \*\*ppOb;*

- задаем число объектов, например, вводим с клавиатуры или читаем из файла как значение целой переменой *n*, и создаем массив указателей динамически *ppOb=new MyClass\*[n];*

- в цикле для каждого объекта вводим параметры или читаем их из файла для его инициализации и создаем объект динамически с помощью вызова конструктора с параметрами, это действие может иметь вид *ppOb[i]=new MyClass(…);* где вместо многоточия передаются фактические параметры для конструктора класса.

При использовании последовательного контейнера (vector, deque или list) объект можно добавлять в конец контейнера с помощью метода push\_back, или сразу создавать контейнер из n объектов и далее инициализировать каждый элемент с помощью метода set.

## **1.5. Вопросы для самоконтроля**

1. Классы и объекты.

2. Статусы доступа полей и методов.

3. Назначение и перегрузка конструкторов.

4. Деструктор класса.

5. Способы создания массива объектов класса переменной длины (динамически через new) и инициализация полей объектов такого массива.

6. Ключевое слово this.

7. Ключевое слово explicit, его применение в конструкторах.

**Варианты заданий для лабораторной работы № 1**

Вариант 1

Класс – сотрудник предприятия. Параметры (поля класса) – ФИО, оклад, надбавка к окладу в %. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию вычисления зарплаты (зарплата = оклад + процентная надбавка от оклада\*оклад). Вывести на печать параметры всех сотрудников и суммарную их зарплату.

Вариант 2

Класс – студент. Параметры (поля класса) – ФИО, массив из m-х оценок за последнюю сессию. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию проверки возможности получения студентом стипендии (все оценки без троек). Вывести на печать всех студентов, получающих стипендию.

Вариант 3

Класс – банковский вклад. Параметры (поля класса): ФИО владельца, текущая сумма, годовая процентная ставка (проценты начисляются ежегодно с капитализацией начисленных процентов с основной суммой). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию расчета суммы на счету через заданное число лет (число лет – параметр функции). Вывести на печать параметры всех вкладов и суммарную сумму на счетах через заданное число лет, которое вводится с клавиатуры.

Вариант 4

Класс – аппаратно- программное средство защиты (СЗ) от несанкционированного доступа (НСД). Параметры (поля класса) – название и номер класса защищенности от НСД (Существует семь классов защищенности от НСД, наивысший 1-ый, самый низкий 7, например, если требуется обеспечить защищенность по 3- му классу, то можно использовать СЗ с классами 1, 2 или 3). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию, проверяющую можно ли это СЗ использовать для заданного класса (номер заданного класса – параметр функции), функцию печати параметров СЗ. Вывести на печать параметры тех СЗ, которые можно использовать для заданного класса защищенности, номер класса защищенности вводится с клавиатуры.

Вариант 5

Класс – криптографический метод защиты информации. Параметры (поля класса) – название и тип (симметричный или несимметричный). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать все методы заданного типа, тип вводится с клавиатуры.

Вариант 6

Класс – криптографический метод защиты информации. Параметры (поля класса) – название и тип (симметричный или несимметричный). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать все методы заданного типа, тип вводится с клавиатуры.

Вариант 7

Класс – прямоугольник. Параметры (поля класса) – длина, ширина. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию вычисления площади. Вывести на печать все параметры прямоугольников, площади которых превышают заданное значение, которое вводится с клавиатуры.

Вариант 8

Класс – автомобиль. Параметры (поля класса) – марка, максимальная скорость (км/ч). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать параметры тех автомобилей, максимальная скорость которых превышает заданное значение, введенное с клавиатуры.

Вариант 9

Класс – квадратное уравнение ах2 + bx + c = 0 (a ≠ 0). Параметры (поля класса) – *a, b, c*. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию расчета дискриминанта. Вывести на печать параметры тех уравнений, которые имеют вещественные корни.

Вариант 10

Класс – квадратное уравнение ах2 + bx + c = 0 (a ≠ 0). Параметры (поля класса) – *a, b, c*. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию расчета дискриминанта. Вывести на печать параметры тех уравнений, которые имеют вещественные корни.

Вариант 11

Класс – полином . Параметры (поля класса): n и массив коэффициентов . Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров, функцию вычисления значения полинома при заданном x (x- параметр функции). Вывести на печать параметры всех полиномов и сумму их значений при заданном значении x, вводимом с клавиатуры.

Вариант 12

Класс - книга. Параметры (поля): автор, название, количество страниц. Статус доступа всех полей private. Класс включает: конструктор, функцию печать параметров, при необходимости функции доступа к полям. Распечатать параметры книги с максимальным количеством страниц.

Вариант 13

Класс – межсетевой экран (МЭ). Параметры (поля класса) – название и номер класс защищенности (Существует пять классов защищенности МЭ, наивысший 1-ый, самый низкий 5, например, если требуется использовать МЭ 3- го класса защищенности, то можно использовать МЭ с классами 1, 2 или 3). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию, проверяющую можно ли МЭ использовать для заданного класса (номер заданного класса – параметр функции), функцию печати параметров МЭ. Вывести на печать параметры тех МЭ, которые можно использовать для заданного класса защищенности, номер класса защищенности вводится с клавиатуры.

Вариант 14

Класс – персональный компьютер (ПК). Параметры (поля класса): название процессора, тактовая частота. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать параметры всех компьютеров в порядке невозрастания тактовой частоты.

Вариант 15

Класс – программа- антивирус. Параметры (поля класса): название, число вредоносных программ в базе. Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать параметры антивирусов и антивирус с самой большой базой вредоносных программ.

Вариант 16

Класс - вещественное число, записанное с точкой. Параметры - его значение (вещественный тип) и запись (строковое представление числа в десятичной системе счисления с точкой). Статус доступа всех полей private. Класс включает: конструктор, при необходимости функции доступа к полям, функцию, определяющую количество цифр в целой части числа в десятичной записи, функцию печати параметров. Напечатать все числа, сумму введенных чисел и суммарное количество цифр в целых частях всех чисел.

Вариант 17

Класс - предложение. Параметры: массив слов (n<10) и их количество. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую количество слов, длиннее 5 букв, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры предложений и процент слов длиннее 5 букв в заданном тексте.

Вариант 18

Класс - выражение, состоящее из целых чисел и знаков операций (скобок нет, операции – сложение, вычитание и умножение).

Параметры: массив значений чисел (n<10), количество чисел и массив знаков операций (тип char). Статус доступа всех полей private. Класс включает: конструктор и функцию, вычисляющую результат, при необходимости функции доступа к полям, функцию печати параметров. Ввести несколько выражений и вывести результаты в порядке, обратном вводу.

Вариант 19

Класс - некоторый товар в магазине. Параметры: наименование, количество и стоимость. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую суммарную стоимость товара, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех товаров и суммарную стоимость всех товаров в магазине.

Вариант 20

Класс - некоторый товар в магазине. Параметры: наименование, количество и закупочная цена. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую стоимость товара исходя из заданного процента прибыли (процент прибыли – параметр функции), при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех товаров и суммарную стоимость всех товаров в магазине с учетом заданного процента прибыли, который вводится с клавиатуры.

Вариант 21

Класс - студент. Параметры (поля): ФИО, массив экзаменационных оценок (*m*=4). Статус доступа всех полей private. Класс включает: конструктор и функцию определения среднего балла, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех студентов и 3-х самых сильных студентов группы.

Вариант 22

Класс - ангар. Параметры (поля): ширина и длина. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую площадь помещения, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры ангаров и площадь склада, состоящего из этих ангаров.

Вариант 23

Класс - квартира. Параметры (поля): общая площадь и стоимость одного квадратного метра. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую стоимость квартиры, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех квартир и все квартиры, стоимость которых не превышает заданной суммы (сумма вводится с клавиатуры).

Вариант 24

Класс - квартира. Параметры (поля): стоимость и количество комнат. Статус доступа всех полей private. Класс включает: конструктор и функцию, определяющую среднюю стоимость одной комнаты, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех квартир и все квартиры, стоимость одной комнаты в которых не превышает заданной суммы (сумма вводится с клавиатуры).

Вариант 25

Класс - выставочные экспонаты. Параметры (поля): название, время экспонирования (в днях), стоимость одного дня экспонирования. Статус доступа всех полей private. Класс включает: конструктор и функцию определения стоимости экспонирования, при необходимости функции доступа к полям, функцию печати параметров. Вывести на печать параметры экспонатов и экспонат, стоимость экспонирования которого максимальна.

Вариант 26

Класс - книга. Параметры (поля): автор, название, количество экземпляров и количество желающих ее прочитать читателей. Статус доступа всех полей private. Класс включает: конструктор и функцию определения средней длины очереди на чтение каждого экземпляра, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры книг и наиболее читаемую книгу в библиотеке.

Вариант 27

Класс – вектор на плоскости. Параметры (поля): координаты конца вектора: x, y (начало вектора в точке с координатами 0, 0), Статус доступа всех полей private. Класс включает: конструктор, функцию печати параметров, при необходимости функции доступа к полям, функцию вычисления длины вектора. Вывести на печать все вектора и вектор с наибольшей длиной.

Вариант 28

Класс - студент. Параметры (поля): ФИО, массив экзаменационных оценок (*m*=4). Статус доступа всех полей private. Класс включает: конструктор и функцию определения среднего балла, при необходимости функции доступа к полям, функцию печати параметров. Напечатать параметры всех студентов и 3-х самых сильных студентов группы.