# Лабораторная работа №5. Шаблоны классов

**1 Цель работы**

Цель работы – ознакомиться с понятием шаблона класса и способами его объявления, изучить методы создания объектов классов, порождаемых шаблонами.

**2 Порядок выполнения работы**

* ознакомиться с описанием лабораторной работы;
* получить задание у преподавателя по вариантам;
* разработать и отладить программу;
* составить и защитить отчет по лабораторной работе у преподавателя.

**3 Содержание отчета**

* титульный лист;
* краткое теоретическое описание;
* задание на лабораторную работу, включающее математическую формулировку задачи;
* результаты выполнения работы, включающие схему алгоритма, тексты программ, результаты вычислений;

**4 Краткая теория**

По аналогии с шаблонами функций можно определять и шаблоны семейства классов:

**template <список\_параметров\_шаблона>**

**определение\_класса**

Шаблон семейства классов определяет способ построения отдельных классов подобно тому, как класс определяет правила построения и формат отдельных объектов. В определении класса, входящего в шаблон, особую роль играет имя класса. Оно является не именем отдельного класса, а параметризованным именем семейства классов.

Определение шаблона может быть только глобальным.

Далее будет приведен пример определения шаблона класса вектора (одномерного массива). Какой бы тип ни имели элементы массива (целый, вещественный, с двойной точностью и т.д.), в этом классе должны быть определены одни и те же базовые операции, например, доступ к элементу по индексу и т.д. если тип элементов вектора задавать как параметр шаблона класса, то система будет формировать вектор нужного типа (и соответствующий класс) при каждом определении конкретного объекта.

Когда шаблон введен, появляется возможность определять конкретные объекты конкретных классов, каждый из которых параметрически порожден из шаблона. Формат определения объекта одного из классов, порождаемых шаблоном классов:

**имя\_параметризованного\_класса <фактические\_параметры\_шаблона>**

**имя\_объекта (параметры\_конструктора)**

В программе шаблон семейства классов с общим именем **Vector** используется для формирования двух классов с массивами целого и символьного типов. В соответствии с требованием синтаксиса имя параметризованного класса, определенное в шаблоне (в примере **Vector**), используется в программе только с последующим конкретным фактическим параметром (аргументом), заключенным в угловые скобки. Параметром может быть имя стандартного или определенного пользователем типа. В данном примере использованы стандартные типы **int** и **char**. Использовать имя **Vector** без указания фактического параметра шаблона нельзя – никакое умалчиваемое значение при этом не предусматривается.

В списке параметров шаблона могут присутствовать формальные параметры, не определяющие тип, точнее – это параметры, для которых тип фиксирован.

**5 Пример программы**

Шаблон класса «вектор (одномерный массив) произвольного типа».

**#include <iostream>**

**using namespace std;**

**template <class T> //Т – параметр шаблона**

**class Vector**

**{**

**T \*data; //указатель на элементы массива**

**int size; //количество элементов в массиве**

**public:**

**Vector(int);**

**~Vector()**

**{**

**delete[] data;**

**}**

**T& operator [](int i) //перегрузка операции []**

**{**

**return data[i];**

**}**

**};**

**template <class T>**

**Vector <T>::Vector(int n)**

**{**

**data = new T[n];**

**size = n;**

**}**

**void main()**

**{**

**Vector <int> X(5); //объект класса «целочисленный вектор»**

**Vector <char> C(5); //объект класса «символьный вектор»**

**for (int i=0;i<5;i++)**

**{**

**X[i] = i;**

**C[i] = 'A'+i;**

**cout<<X[i]<<' '<<C[i]<<'\t';**

**}**

**}**

Результат выполнения программы:

**0 A 1 B 2 C 3 D 4 E**

**6 Контрольные вопросы**

1. Для чего используются шаблоны классов? Что у них общего с шаблонами функций?
2. Как описываются шаблоны классов?
3. Как создать объект на основе класса, порожденного шаблоном?
4. Каких типов могут быть фактические параметры шаблонов классов?
5. Можно ли описывать в списке параметров шаблона параметры, не определяющие тип?

**7 Варианты заданий для самостоятельного решения**

Для каждого варианта необходимо разработать шаблон соответствующего класса, где поля могут иметь различные типы данных (некоторые поля могут быть статическими). Предусмотреть наличие в классе указанных методов и перегруженных операций. Варианты заданий приведены в таблице 5.1.

*Таблица 5.1 – Варианты заданий*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название класса | Поля | Методы | Перегружаемые операции |
|  | Квадрат | Координаты центра, длина стороны | Нарисовать, определить периметр и площадь | ==, += (увеличить длину стороны) |
|  | Круг | Координаты центра, радиус | Нарисовать, определить периметр и площадь | > (сравнение по радиусам), постфиксный –– (уменьшить радиус) |
|  | Стек (на массиве) | Указатель на вершину, текущий размер | Добавить, извлечь элемент, проверить на пустоту, распечатать | += (поместить в стек), префиксный – – (извлечь из стека) |
|  | Очередь (на массиве) | Указатели на начало и конец, текущий размер | Добавить, извлечь элемент, проверить на пустоту, распечатать | += (поместить в очередь), постфиксный – – (извлечь из очереди) |
|  | Односвязный список | Указатели на начало и конец списка, текущий размер | Добавить элемент в начало, удалить из начала, найти элемент | << (распечатать весь список), == (равны ли размеры двух списков) |
|  | Двусвязный список | Указатели начала и конца, текущий размер | Добавить элемент в конец, удалить из конца, распечатать список | << (распечатать весь список), == (равны ли размеры двух списков) |
|  | Односвязный список | Указатели начала и конца, информационное поле элемента, ссылка на следующий элемент | Добавить элемент в конец, удалить из конца, распечатать список | [] (получить элемент списка по его номеру), постфиксный ++ (увеличить на 1 инф. поле каждого элемента) |
|  | Точка | Координаты, цвет | Нарисовать, стереть, переместить | Бинарный – (расстояние между точками), –= (уменьшить каждую координату) |
|  | Двусвязный список | Указатели начала и конца, текущий размер | Добавить элемент в начало, удалить из начала, найти элемент | [] (получить элемент списка по его номеру), префиксный – – (уменьшить на 1 информационное поле каждого элемента) |
|  | Эллипс | Координаты, 2 радиуса | Нарисовать, стереть, переместить | !=, префиксный ++ (увеличить каждую координату) |
|  | Комплексное число | Действительная и мнимая части | Изменить значение действительной и мнимой частей, определить модуль | Унарный – (поменять знак каждой части), бинарный + |
|  | Квадратное уравнение | Коэффициенты, дискриминант, корни | Вывести на экран уравнение, решить уравнение, вывести корни | Бинарный –, \*= (каждый коэффициент умножить на число) |
|  | Арифметическое выражение | 2 операнда, знак операции | Вывести на экран выражение, вычислить, вывести результат | ==, постфиксный –– (уменьшить каждый операнд) |
|  | Матрица | 2 размерности, указатель на элементы | Вывести на экран, найти максимум и среднее значение элементов | + (сложение двух матриц), < (каждый элемент 1-й матрицы меньше соотв. элемента второй) |
|  | Дробь | Целая часть, числитель, знаменатель | Изменить значения полей, сократить, вывести на экран | << (вывести в виде десятичной дроби), \* (перемножить две дроби) |
|  | Дата | Число, месяц, год | Изменить значения полей, определить, сколько дней осталось до нового года, вывести в формате «чч/мм/гггг» | != (проверка двух дат на неравенство), бинарный – (промежуток между двумя датами в днях, месяцах, годах) |
|  | Время | Часы, минуты, секунды | Изменить значения полей, вывести в формате «чч:мм:сс», определить, является ли заданное время «до полудня» или «после полудня» | >> (ввод новых значений полей), += (добавить заданное количество секунд) |
|  | Отрезок | Координаты концов | Изменить значения полей, вычислить длину, проверка на вырождение отрезка в точку | << (нарисовать), == (проверка на равенство длин двух отрезков) |
|  | Многочлен | Порядок многочлена, массив коэффициентов | Вывести на экран, изменить значения коэффициентов, найти производную | +, [] (получить заданный индексом коэффициент) |
|  | Треугольник | Длины сторон, тип | Проверка возможности построения треугольника, определение типа (если нельзя построить треугольник, то тип = «no»), вычисление площади | > (сравнение площадей двух треугольников), постфиксный – – (декремент каждой стороны) |
|  | Множество | Элементы множества, текущее количество элементов | Проверка на включение элемента в множество, вывод на экран, удаление элемента из множества | += (добавление элемента в множество), \* (пересечение множеств) |
|  | Линейное уравнение | Коэффициенты, корень | Изменение значений полей, вывод в формате (ax+b=0), решение уравнения | <= (сравнение корней двух уравнений), префиксный ++ (инкремент каждого коэффициента) |
|  | Граф | Количество вершин, матрица инцидентности (или смежности) | Определение количества ребер, поиск кратчайшего пути из одной вершины в другую, обход графа в глубину | [] (определение количества вершин, смежных с заданной), унарный – (изменение направления всех дуг графа) |
|  | Узел бинарного дерева | Указатель на корень, информационное поле узла, ссылки на левого и правого потомков | Распечатать дерево, определить среднее арифметическое элементов, найти и удалить заданный элемент | << (обход дерева сверху вниз), += (добавить элемент в дерево) |
|  | Правильная треугольная пирамида | Стороны основания, высота | Изменить значения полей, вычислить площадь поверхности, вычислить объем | << (вывод площади основания), > (сравнение высот двух пирамид) |
|  | Цилиндр | Радиус основания, высота | Изменить значения полей, вычислить площадь поверхности, вычислить объем | == (проверка на равенство по всем полям), префиксный – – (декремент высоты) |
|  | Ромб | Координаты центра, длины диагоналей | Нарисовать, вычислить периметр, вычислить площадь | Постфиксный ++ (инкремент каждой диагонали), –= (смещение координат центра) |
|  | Биквадратное уравнение | Коэффициенты, корни | Изменение значений полей, вывод в формате (ax^4+bx^2+c=0), решение уравнения | >> (ввод значений коэффициентов), бинарный – (разность двух уравнений) |