Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

Лабораторная работа №7.

" Шаблоны классов "

15 вариант

Выполнил студент гр. РИС-24-2б

Иванова Елена Олеговна

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

Цель задания:

1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
2. Реализация шаблона класса-контейнера.

Постановка задачи

1. Определить шаблон класса-контейнера (см. лабораторную работу №6).
2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.
3. Перегрузить операции, указанные в варианте.
4. Инстанцировать шаблон для стандартных типов данных (int, float, double).
5. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы стандартных типов данных.
6. Реализовать пользовательский класс (см. лабораторную работу №3).
7. Перегрузить для пользовательского класса операции ввода-вывода.
8. Перегрузить операции необходимые для выполнения операций контейнерного класса.
9. Инстанцировать шаблон для пользовательского класса.
10. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы пользовательского класса.

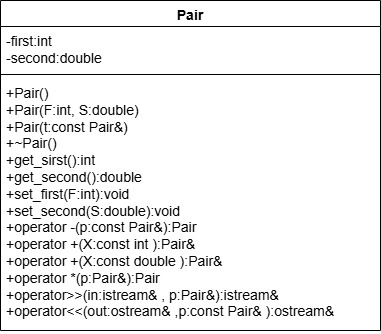
Задача.

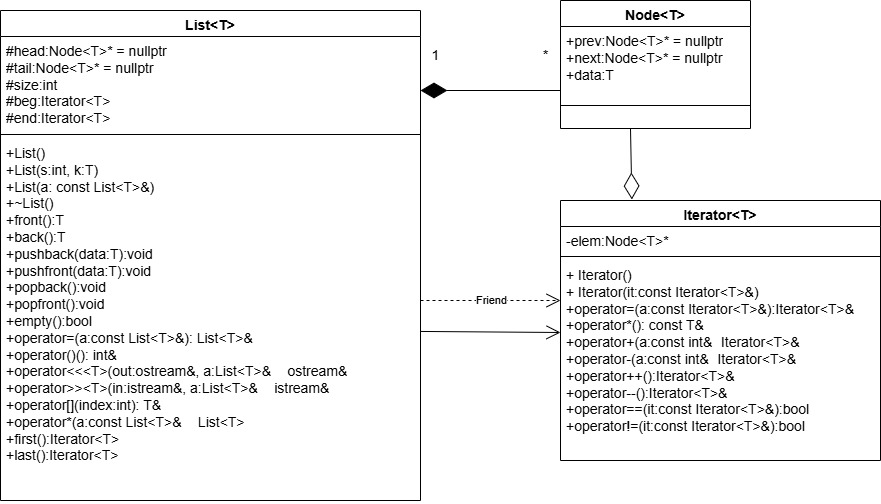
Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции: [] – доступа по индексу; int() – определение размера списка; \* вектор – умножение элементов списков a[i]\*b[i]; Пользовательский класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

Анализ задачи.

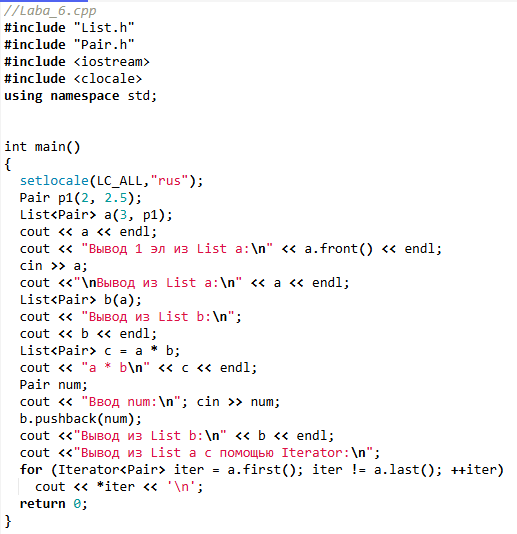
1. Взять шаблон класса контейнера из 6 лабы.
2. Взять класс из 3 лабы.
3. Инстанцировать шаблон для пользовательского класса.

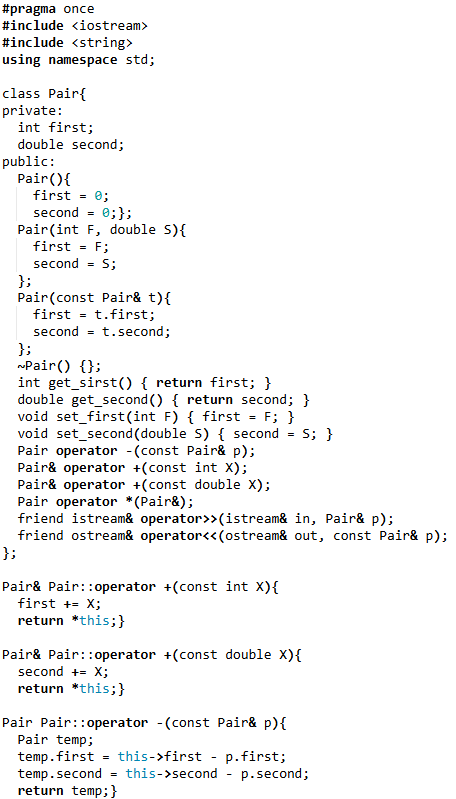
Диаграмма.

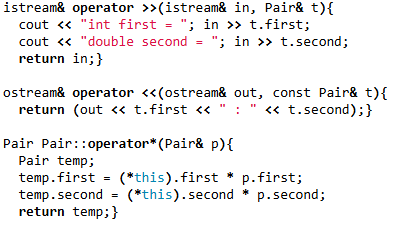


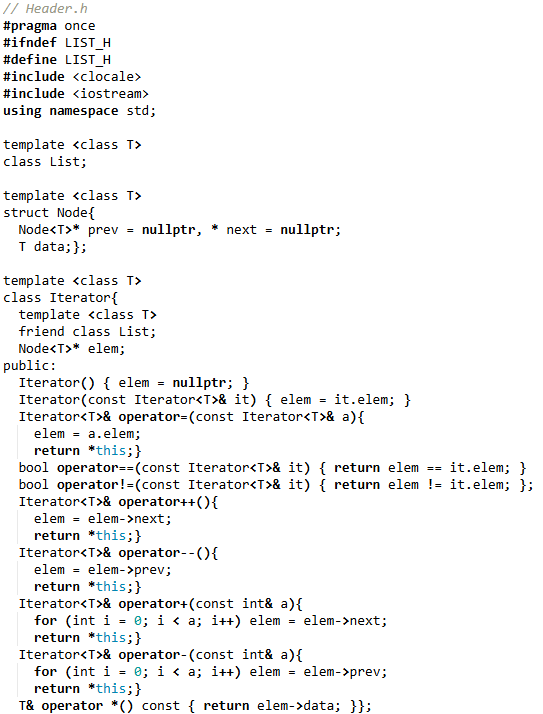


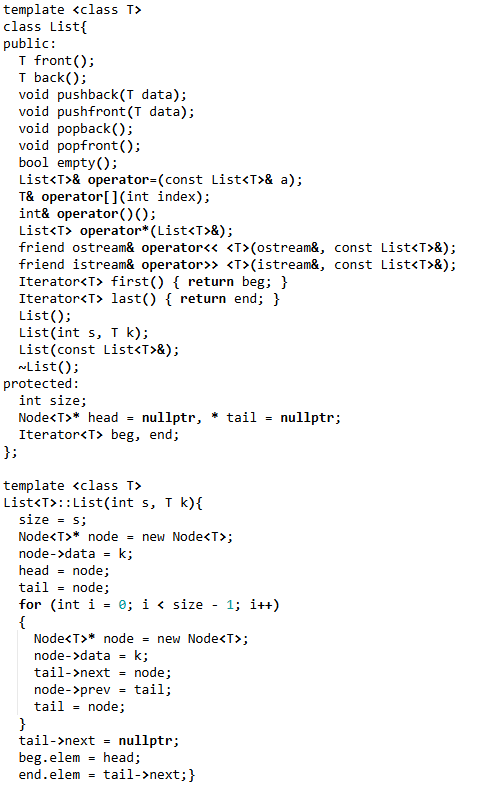
Код.

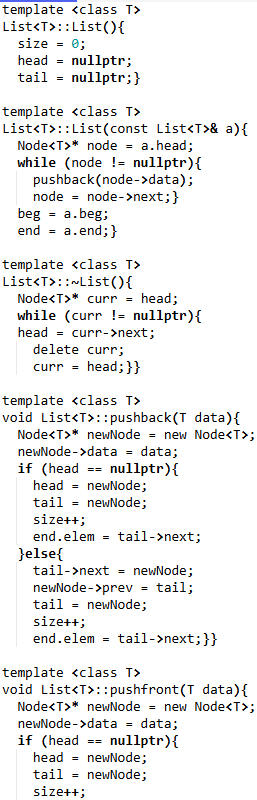


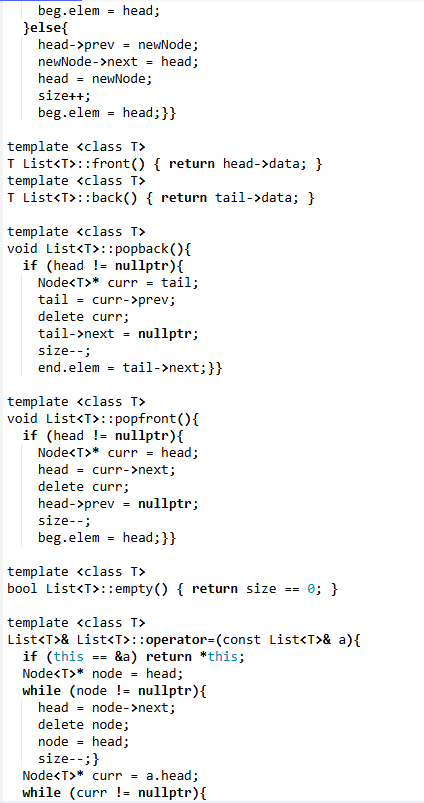


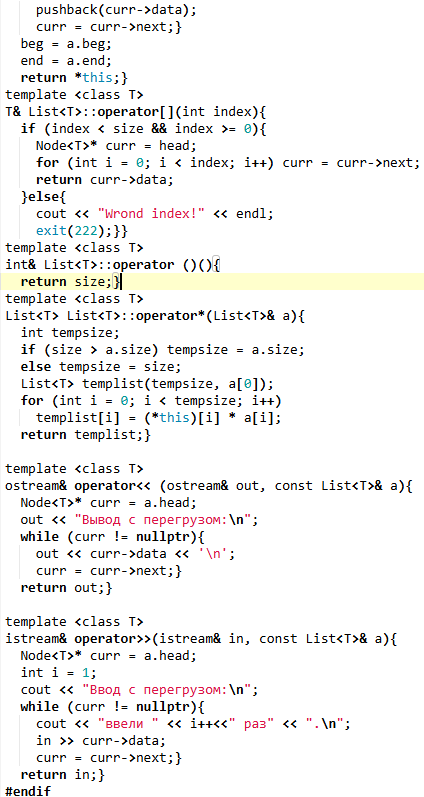




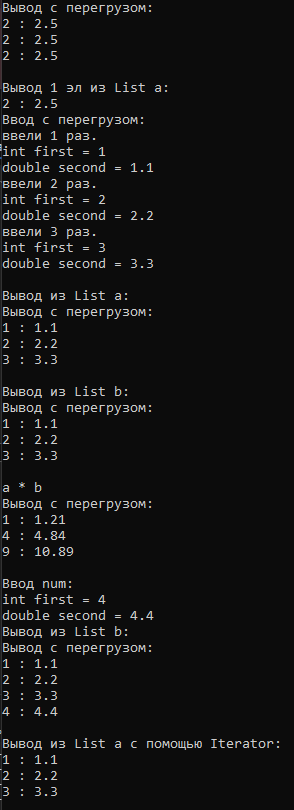


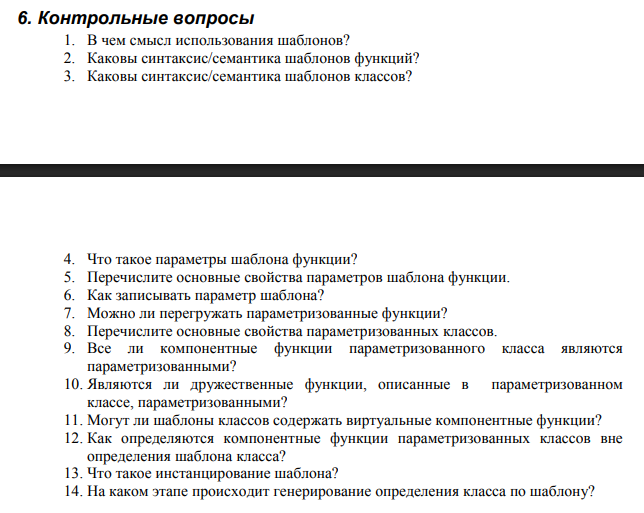






Вывод.





Контрольные вопросы.

1. Шаблоны вводятся для того, чтобы автоматизировать создание функций, обрабатывающих разнотипные данные.
2. template<параметры\_шаблона>

заголовок функции

{тело функции}

пример:

template<class type>

type abs(type x)

{

if (x<0) return -x;

else return x;

}

1. template<параметры\_шаблона>

class имя\_класса

{ ... };

пример:

template<class T>

class Point{

T x, y;

public:

Point(T x = 0, T y = 0):x(x), y(y){}

void Show();{;

1. Параметр шаблона - то, что указывается в <>.

Параметрами шаблонов могут быть:

⦁ параметры-типы;

⦁ параметры обычных типов;

⦁ параметры-шаблоны.

1. - шаблоны не могут быть виртуальными

- шаблоны могут содержать статические элементы, дружественные функции и классы

- шаблоны могут быть производными как от шаблонов, так и от обычных классов, а также являться базовыми и для шаблонов, и для обычных классов.

1. в угловых скобочках < >
2. Да, меняя тип данных параметра, либо меняя параметры местами, в том случае, если они разного типа, а также можно выполнять перегрузку функции добавляя или исключая параметры.
3. - Параметры шаблона могут быть любыми типами данных, включая примитивные типы, пользовательские типы и другие шаблоны классов.

- Параметры шаблона могут быть как значениями, так и типами. Например, можно создать шаблон класса с параметром, который задает размер массива.

-Параметры шаблона могут использоваться внутри класса для объявления переменных, функций и типов.

-Каждый конкретный экземпляр параметризованного класса является отдельным типом, который определяется параметрами шаблона.

-Шаблонные классы могут быть унаследованы от других шаблонных классов и могут иметь шаблонные функции-члены.

1. Да, все компонентные функции (методы) параметризированного класса являются параметризованными, то есть они используют те же параметры шаблона, что и сам класс.
2. Да, дружественные функции, описанные внутри класса, являются параметризованными, если класс является параметризованным. Дружественные функции имеют доступ к закрытым членам класса, но они также должны быть параметризованы, чтобы иметь доступ к параметрам шаблона класса.
3. Да, шаблоны классов могут содержать виртуальные компонентные функции. Виртуальные функции позволяют реализовать полиморфизм и позволяют вызывать функцию на объекте, указатель на который имеет тип базового класса, но указывает на объект, созданный из класса-наследника.
4. Компонентные функции параметризованных классов могут быть определены вне определения шаблона класса с помощью использования template<class T> перед каждым определением функции, где T - это параметр шаблона.
5. Инстанцирование шаблона - это процесс создания конкретной реализации шаблона класса или функции на основе заданных параметров шаблона. При инстанцировании компилятор создает отдельный экземпляр шаблона для конкретных аргументов шаблона. То есть компилятор подставляет значения параметров шаблона в шаблонный код и создает реализацию класса или функции, которая может быть использована в программе.
6. На этапе компиляции.