ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №7

Шаблоны классов

Вариант №15

Выполнил студент группы РИС-23-3Б

Блинов А. Е.

Проверила доцент кафедры ИТАС

О. А. Полякова

2024 г.

**Постановка задачи**

1. Определить шаблон класса-контейнера.

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.

7. Перегрузить для пользовательского класса операции ввода-вывода.

8. Перегрузить операции необходимые для выполнения операций контейнерного класса.

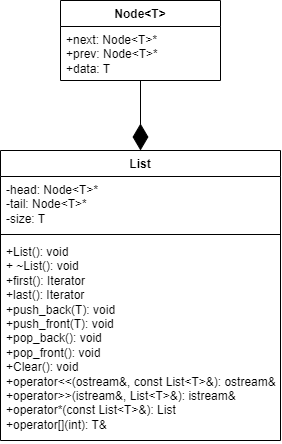
9. Инстанцировать шаблон для пользовательского класса.

10. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы пользовательского класса.

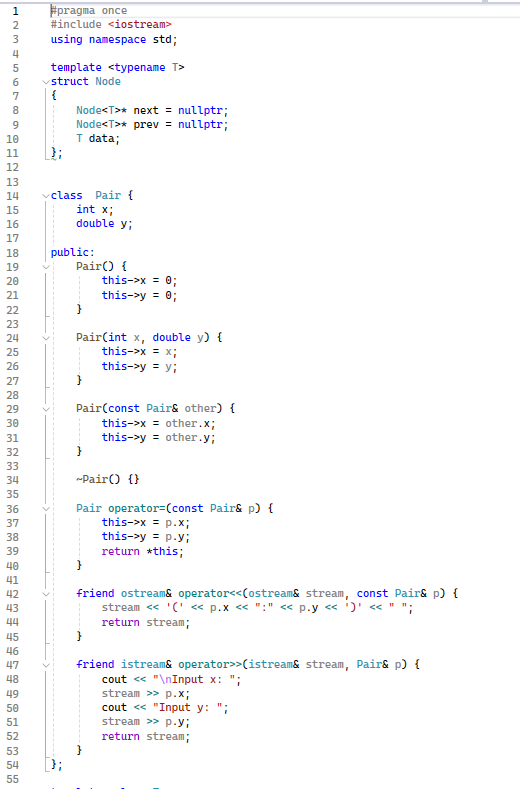
Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции: [] – доступа по индексу; int() – определение размера списка; \* вектор – умножение элементов списков a[i]\*b[i];

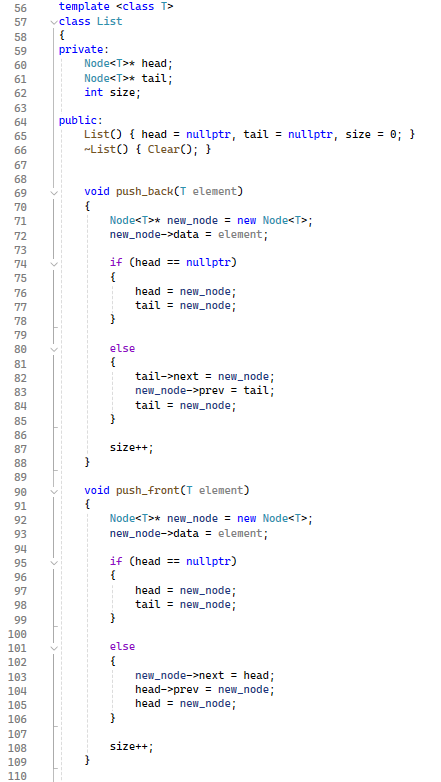
Пользовательский класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

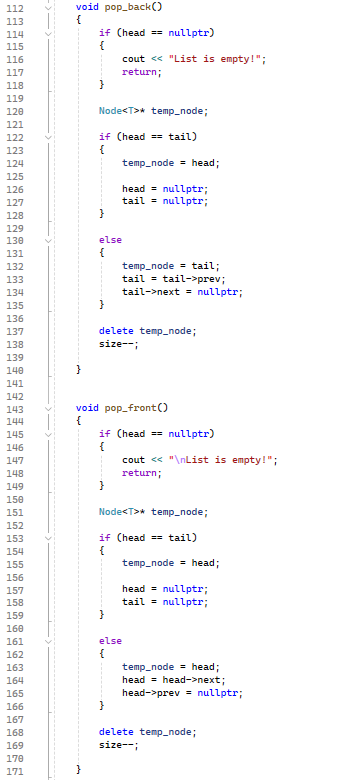
**UML диаграмма**

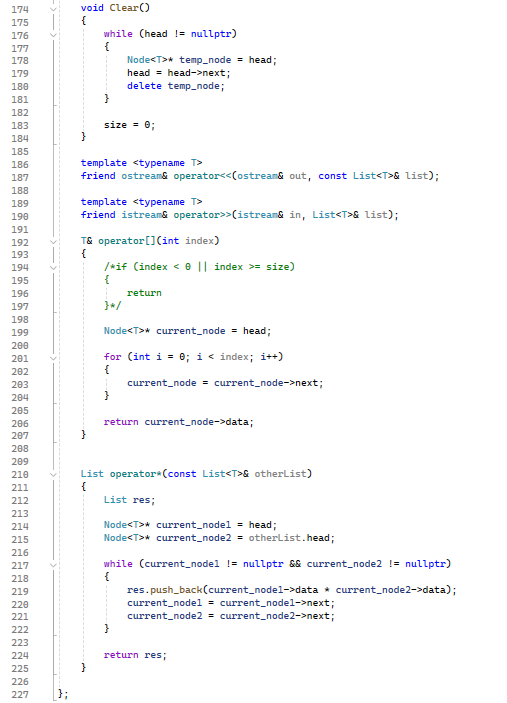


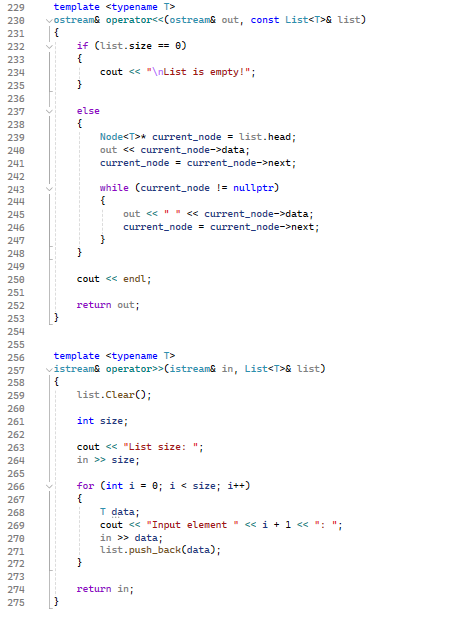
**Код программы**

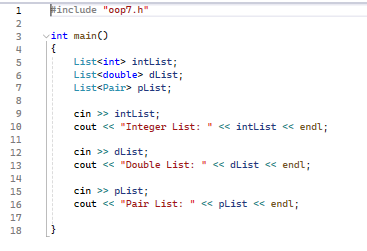




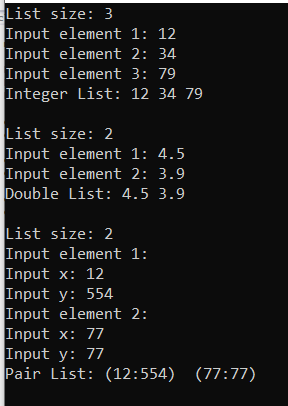








**Результаты**

****

**Вывод**

Программа работает успешно

GitHub: <https://github.com/DimetriusAsti/OOP7.git>

**Контрольные вопросы**

**1. В чем смысл использования шаблонов?**

Использование шаблонов в C++ позволяет писать обобщенный код, который работает с различными типами данных без необходимости повторного написания кода для каждого типа.

**2. Каковы синтаксис/семантика шаблонов функций?**

template <class T>

T functionName(T parameter) {

// код функции

}

**3. Каковы синтаксис/семантика шаблонов классов?**

template <class T>

class ClassName {

// компоненты класса

};

**4. Что такое параметры шаблона функции?**

Параметры шаблона функции – это аргументы, которые передаются шаблону для создания конкретной версии функции.

**5. Перечислите основные свойства параметров шаблона функции**

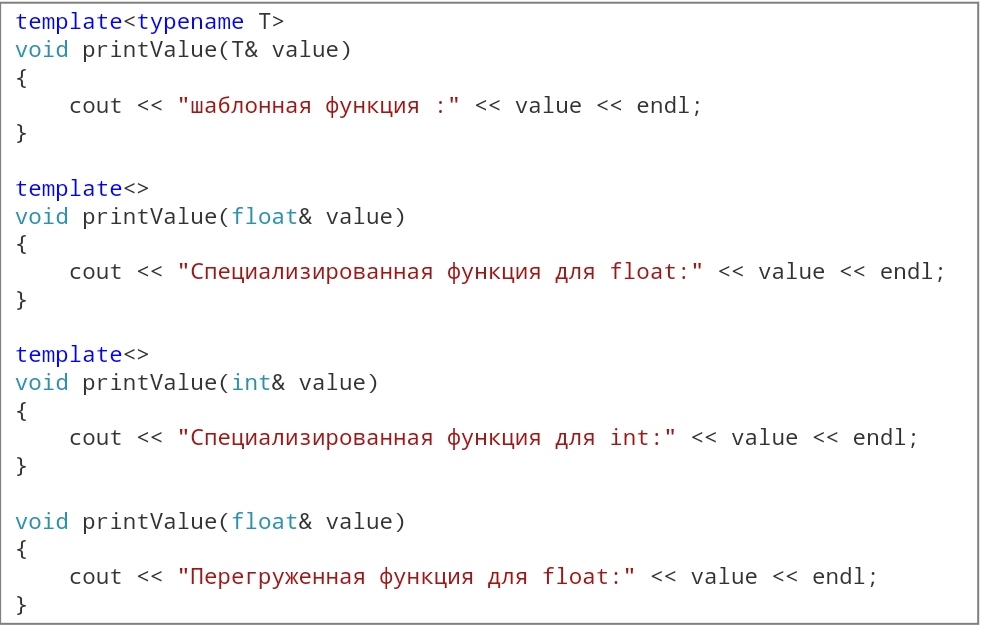
* В списке параметров шаблона может быть несколько параметров, каждому предшествует ключевое слово *typename* или *class*.
* Параметризированная функция может иметь сколь угодно непараметризированных формальных параметров.
* В списке описания прототипа шаблона имена параметров не обязаны совпадать с именами в описании шаблона.
* При использовании шаблонов возможна перегрузка как шаблонов, так и функций

**6. Как записывать параметр шаблона?**

Параметр шаблона записывается как template <class T>, где T - это имя параметра шаблона.

**7. Можно ли перегружать параметризованные функции?**

Да, функции могут быть перегружены даже если они параметризованы шаблоном.



**8. Перечислите основные свойства параметризованных классов.**

1. Компонентные функции параметризованного класса автоматически являются параметризованными.
2. В параметризованном классе нельзя определить дружественные параметризованные классы
3. Шаблоны функций-членов нельзя описывать как виртуальные.
4. Локальные (вложенные) классы не могут содержать шаблоны в качестве своих элементов
5. Дружественные функции, которые описываются в параметризованном классе, не являются автоматически параметризованными, т.е. по умолчанию такие функции являются дружественными для всех классов, которые организуются по шаблону.

**9. Все ли компонентные функции параметризованного класса являются параметризованными?**

Нет, компонентные функции параметризованного класса могут быть не параметризованными.

**10. Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?**

Дружественные функции, описанные в параметризованном классе, не являются параметризованными, они просто используют типы, указанные в параметрах шаблона.

**11. Могут ли шаблоны классов содержать виртуальные компонентные функции?**

Да, шаблоны классов могут содержать виртуальные компонентные функции.

**12. Как определяются компонентные функции параметризованных классов вне определения шаблона класса?**

Если компонентная функция описывается вне шаблона класса:

**template <список\_параметров> тип\_функции имя\_класса <список\_имен\_параметров> :: имя\_функции(список параметров) {тело функции}**

**13. Что такое инстанцирование шаблона?**

Инстанцирование шаблона это процесс создания конкретной версии шаблонного класса или функции для конкретного типа данных.

**14. На каком этапе происходит генерирование определения класса по шаблону?**

Генерирование определения класса по шаблону происходит при вызове или использовании шаблонного класса или функции с конкретными типами данных (инстанцирование).