Пермский национальный исследовательский политехнический университет

Кафедра «Информационные технологии и автоматизированные системы»

Лабораторная работа 11

**Последовательные контейнеры библиотеки STL**

Выполнил

Студент гр. РИС-24-3б Рунтов Иван Андреевич

Проверил

Доцент кафедры ИТАС Ольга Андреевна Полякова

г. Пермь, 2025

**Постановка задачи**

**Задача 1.**

1. Создать последовательный контейнер.

2. Заполнить его элементами стандартного типа (тип указан в варианте).

3. Добавить элементы в соответствии с заданием

4. Удалить элементы в соответствии с заданием.

5. Выполнить задание варианта для полученного контейнера.

6. Выполнение всех заданий оформить в виде глобальных функций.

**Задача 2.**

1. Создать последовательный контейнер.

2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.

3. Добавить элементы в соответствии с заданием

4. Удалить элементы в соответствии с заданием.

5. Выполнить задание варианта для полученного контейнера.

6. Выполнение всех заданий оформить в виде глобальных функций.

**Задача 3**

1. Создать параметризированный класс, используя в качестве контейнера последовательный контейнер.

2. Заполнить его элементами.

3. Добавить элементы в соответствии с заданием

4. Удалить элементы в соответствии с заданием.

5. Выполнить задание варианта для полученного контейнера.

6. Выполнение всех заданий оформить в виде методов параметризированного класса.

**Задача 4**

1. Создать адаптер контейнера.

2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.

3. Добавить элементы в соответствии с заданием

4. Удалить элементы в соответствии с заданием.

5. Выполнить задание варианта для полученного контейнера.

6. Выполнение всех заданий оформить в виде глобальных функций.

**Задача 5**

1. Создать параметризированный класс, используя в качестве контейнера адаптер контейнера.

2. Заполнить его элементами.

3. Добавить элементы в соответствии с заданием

4. Удалить элементы в соответствии с заданием.

5. Выполнить задание варианта для полученного контейнера.

6. Выполнение всех заданий оформить в виде методов параметризированного класса.

**Вариант 15**

**Задача 1**

1. Контейнер – список

2. Тип элементов – double

**Задача 2**

Тип элементов Pair (см. лабораторную работу №3).

**Задача 3**

Параметризированный класс – Список (см. лабораторную работу №7)

**Задача 4**

Адаптер контейнера – очередь с приоритетами.

**Задача 5**

Параметризированный класс – Список

Адаптер контейнера – очередь с приоритетами

**Задание 3**

Найти среднее арифметическое и добавить его в конец контейнера

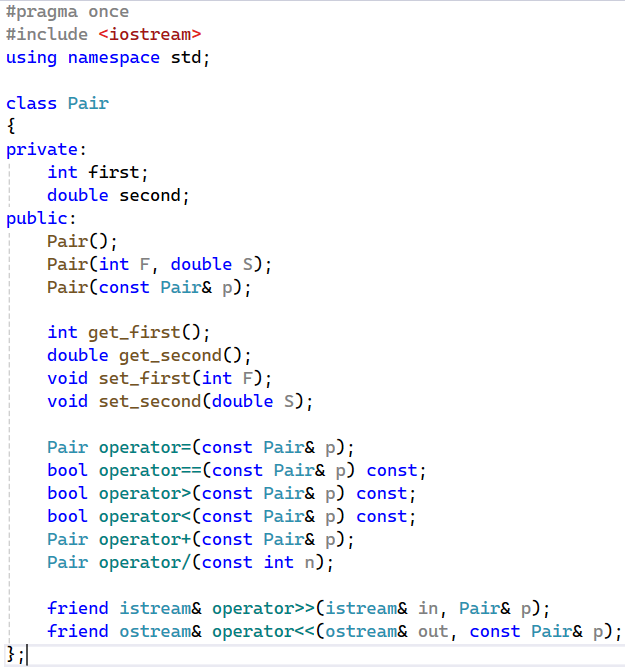
**Задание 4**

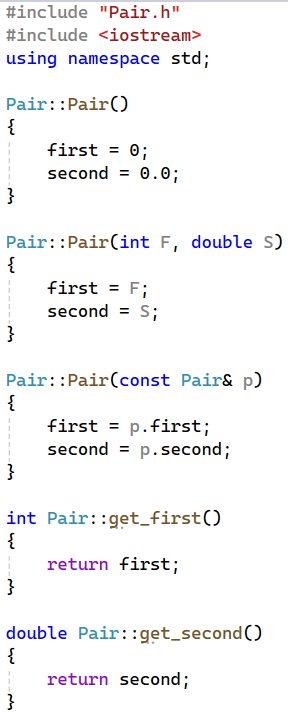
Найти элементы ключами из заданного диапазона и удалить их из контейнера

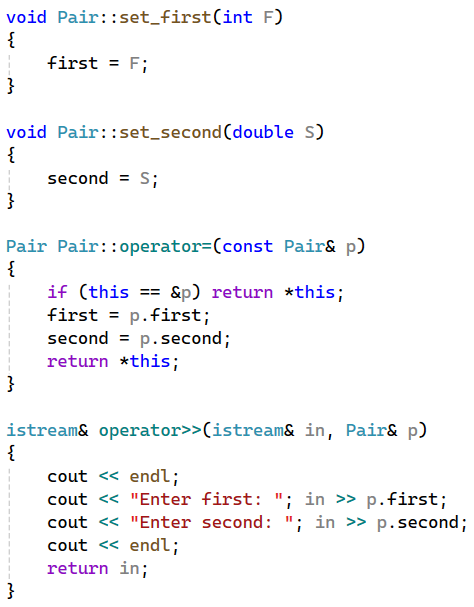
**Задание 5**

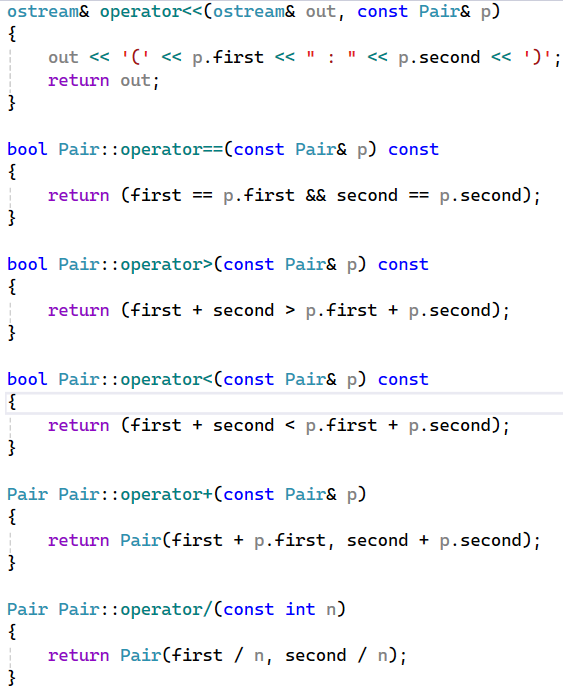
Найти элементы ключами из заданного диапазона и удалить их из контейнера

**Класс Pair**

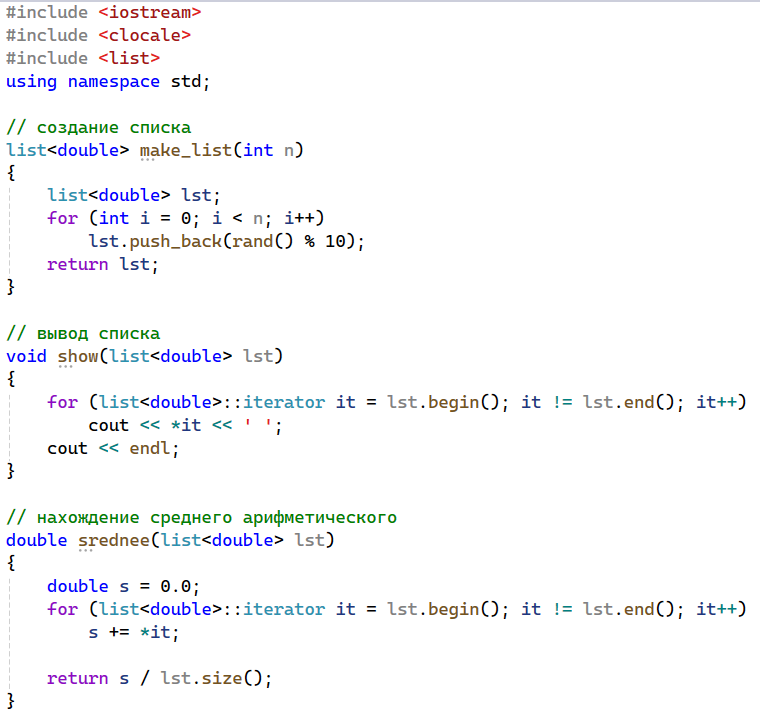


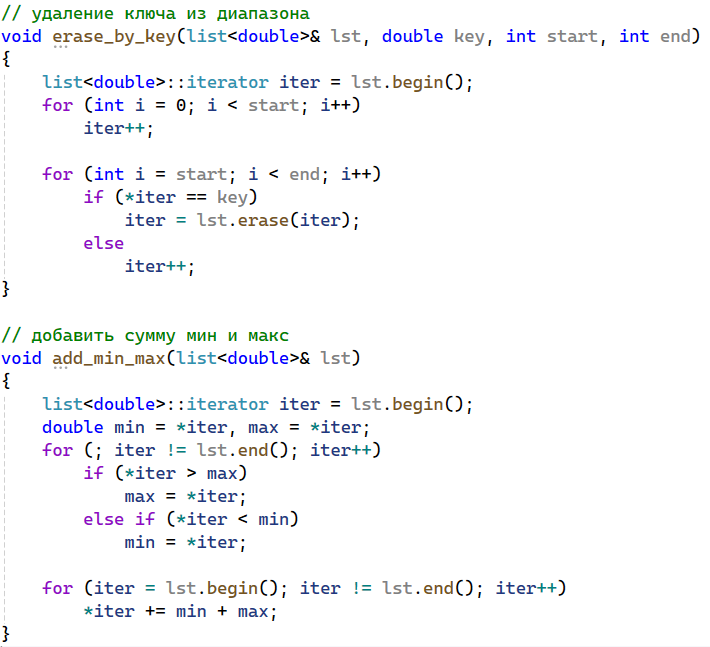


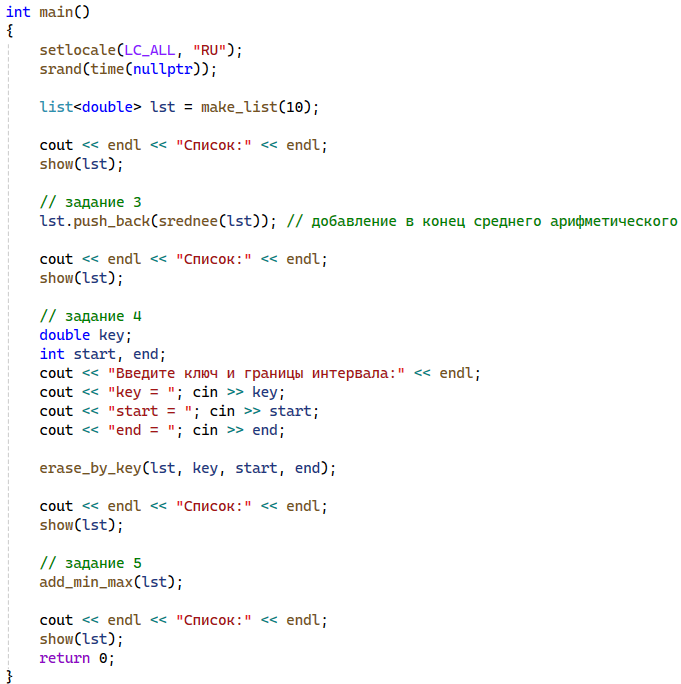




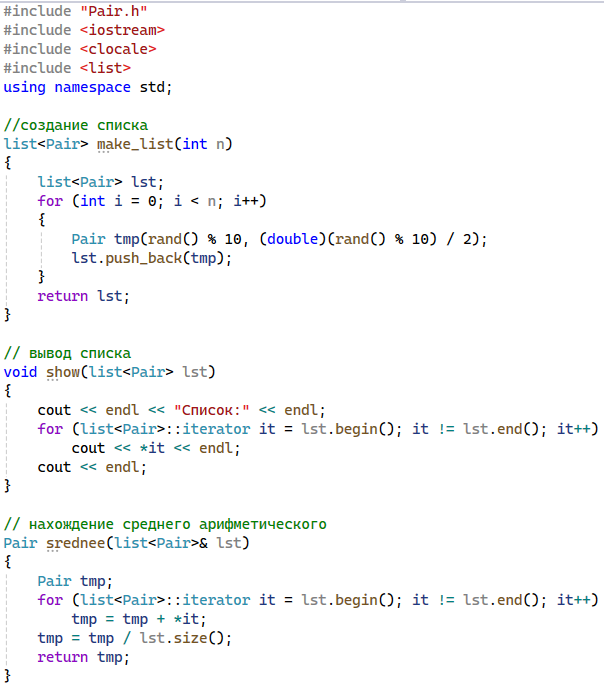
**Задача 1**

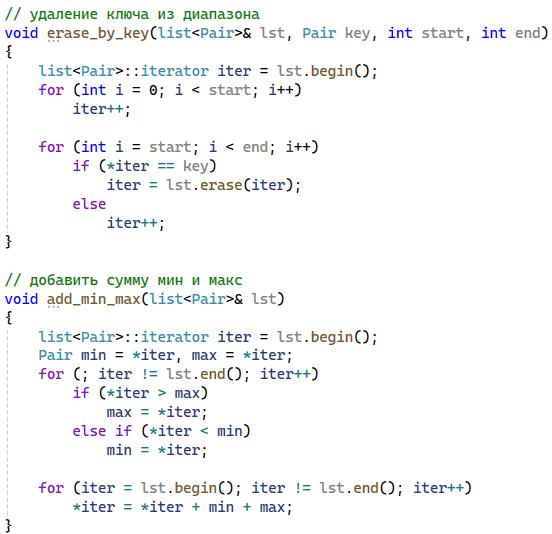


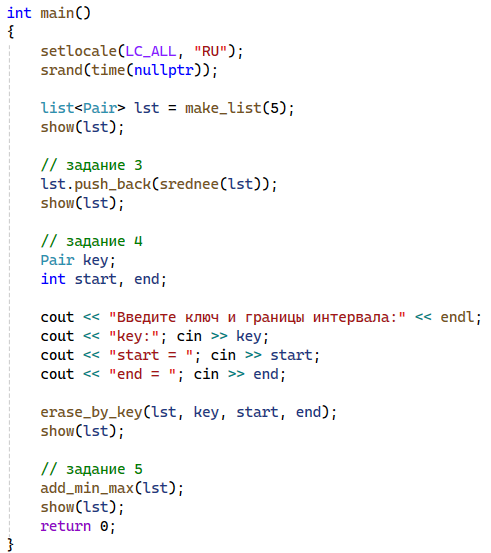




**Задача 2**

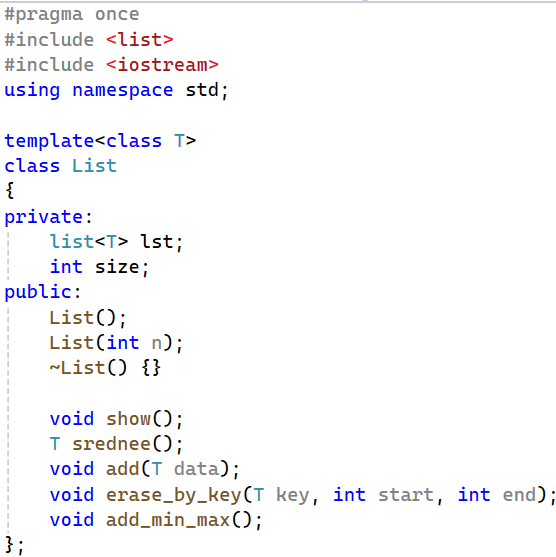


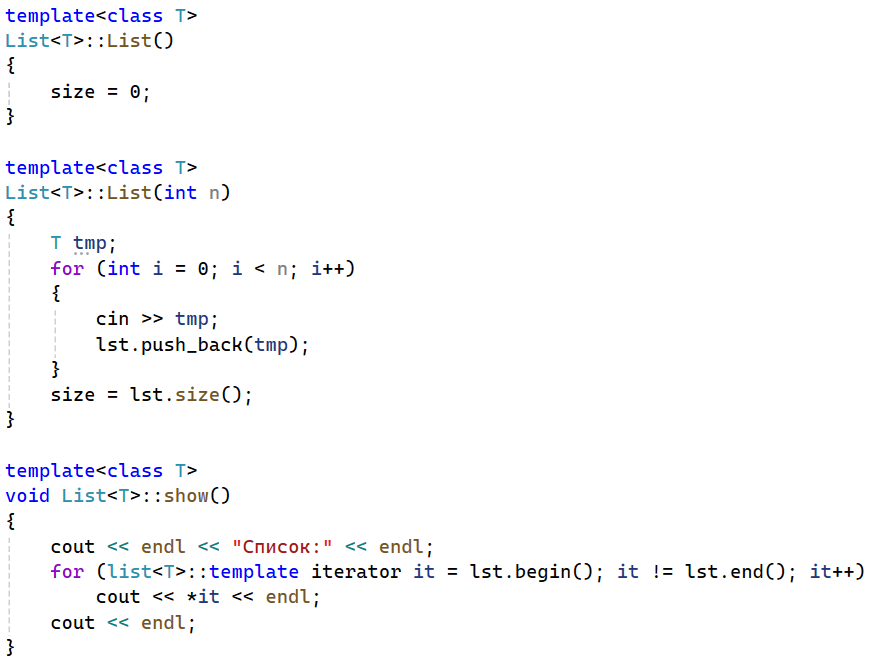


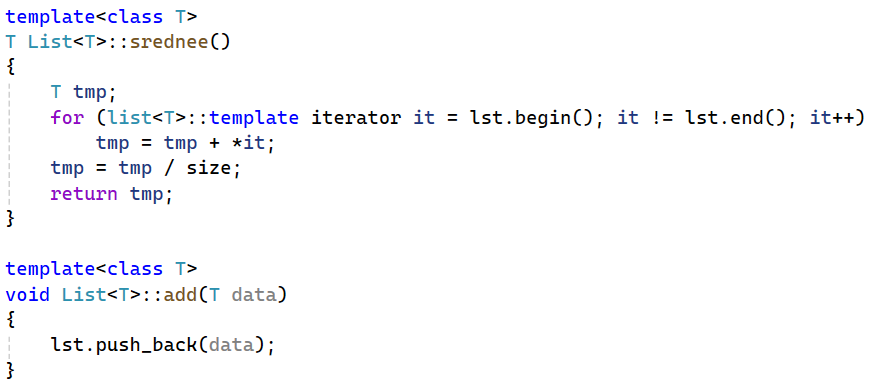


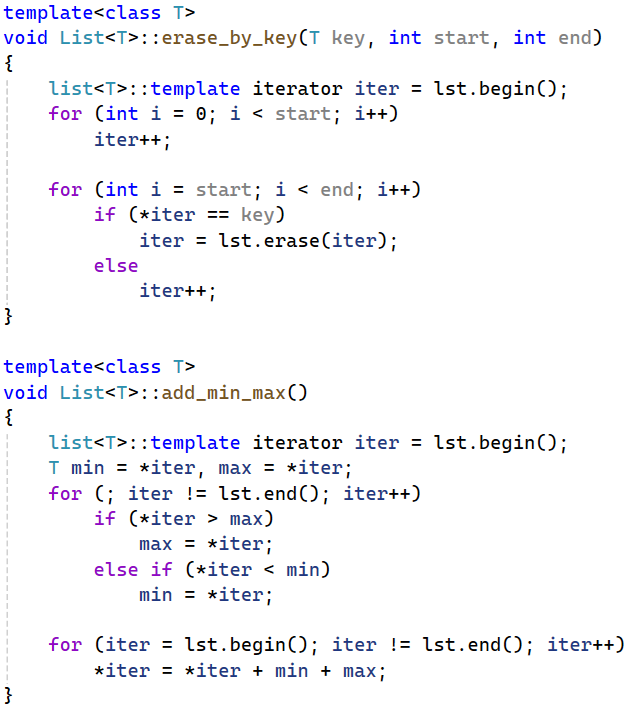
**Задача 3**

**Параметризованный класс List**

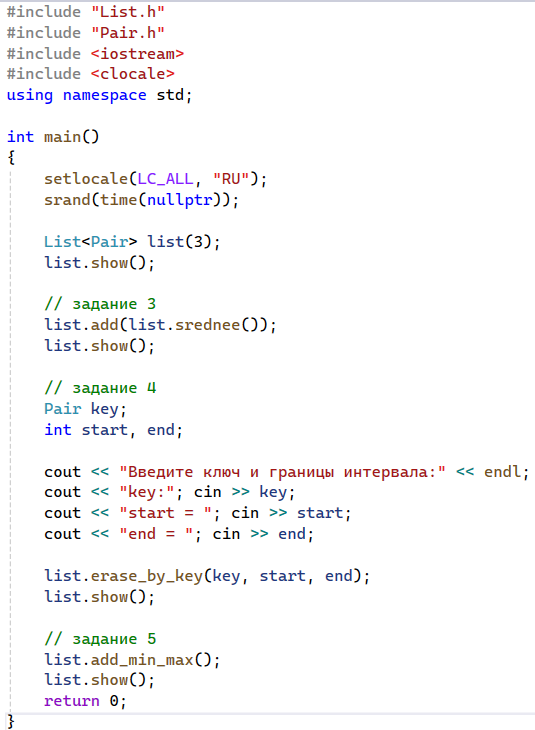




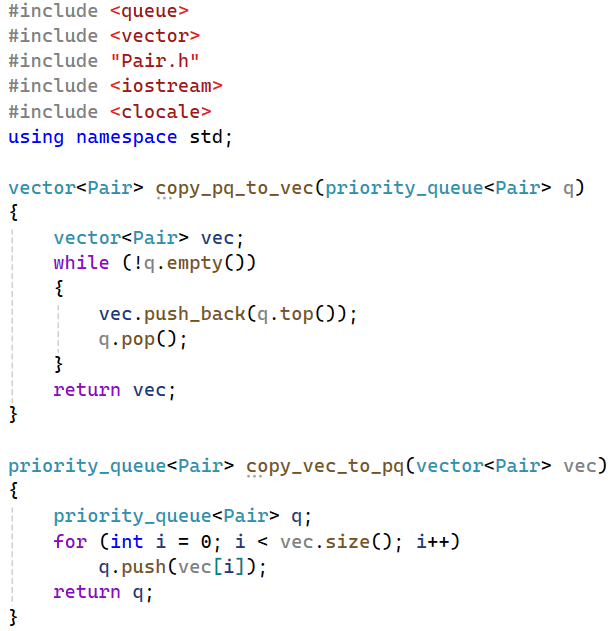


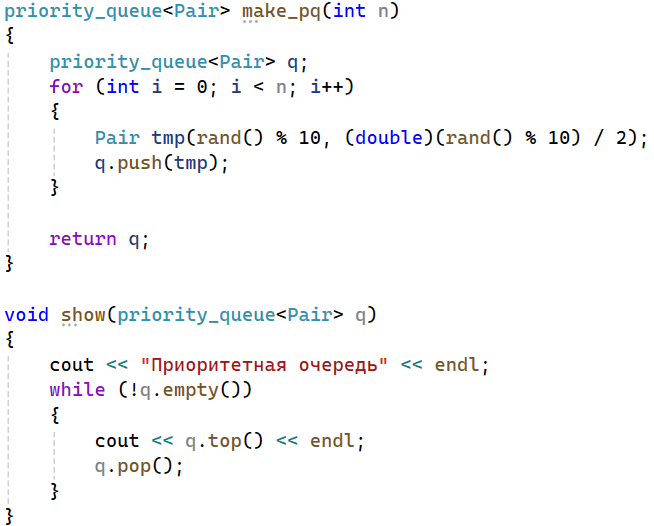


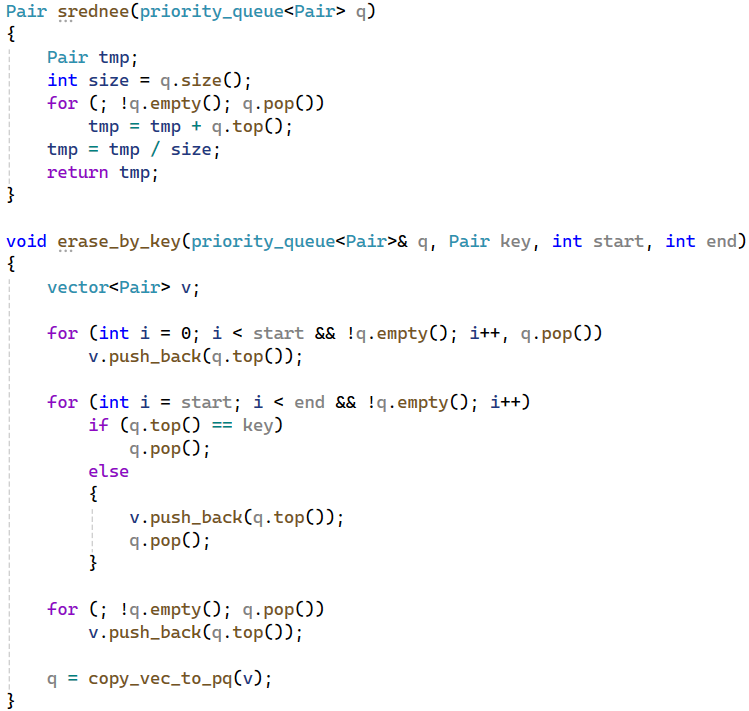
**Основная функция**

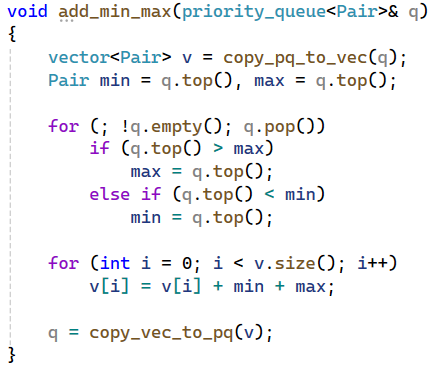


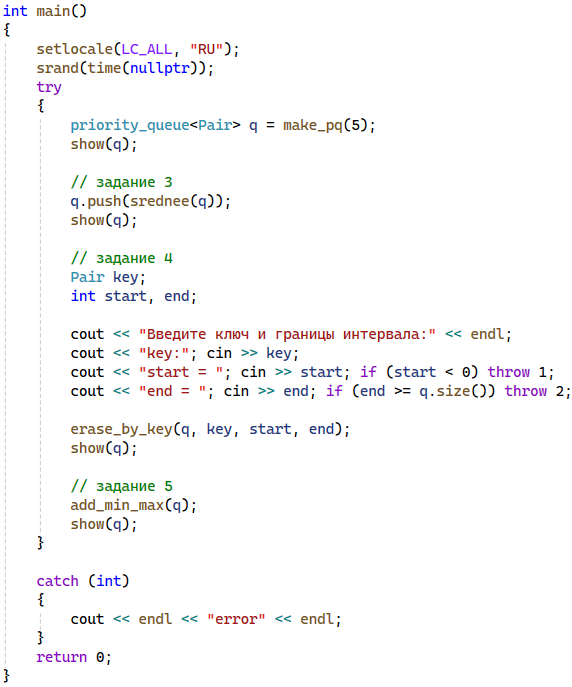
**Задача 4**





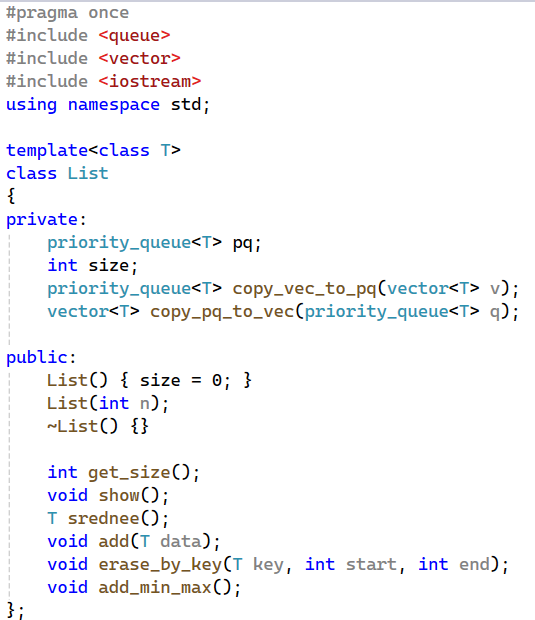


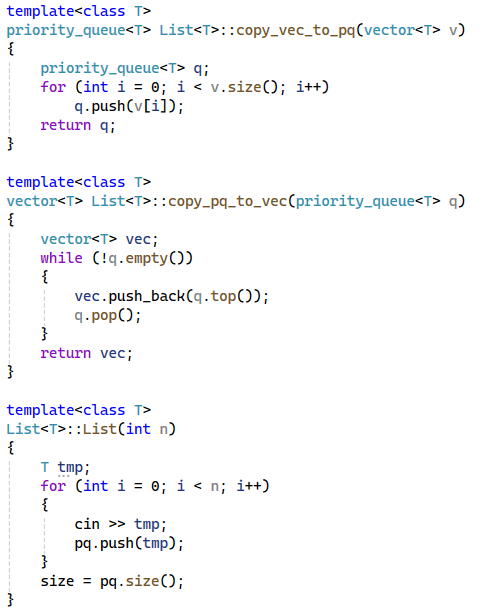


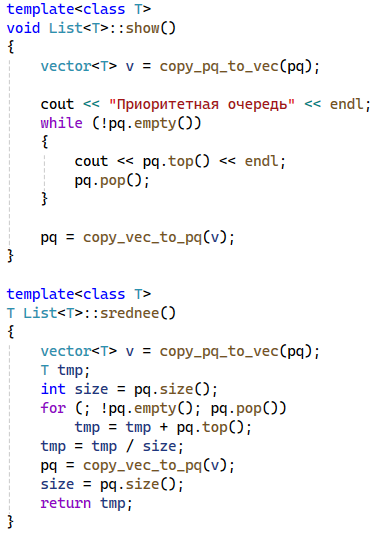


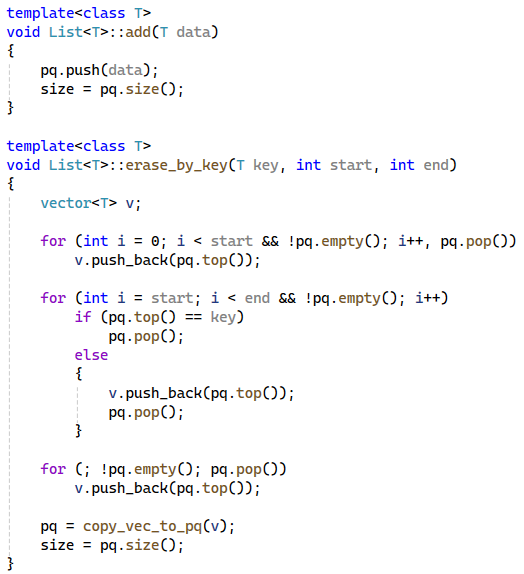
**Задача 5**

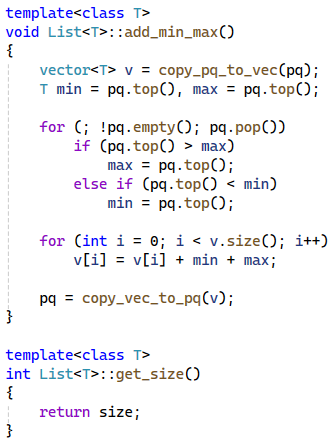
**Параметризованный класс List**



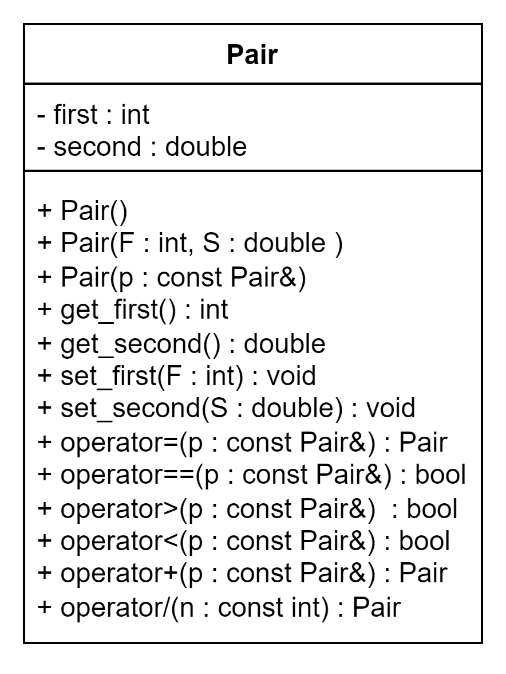


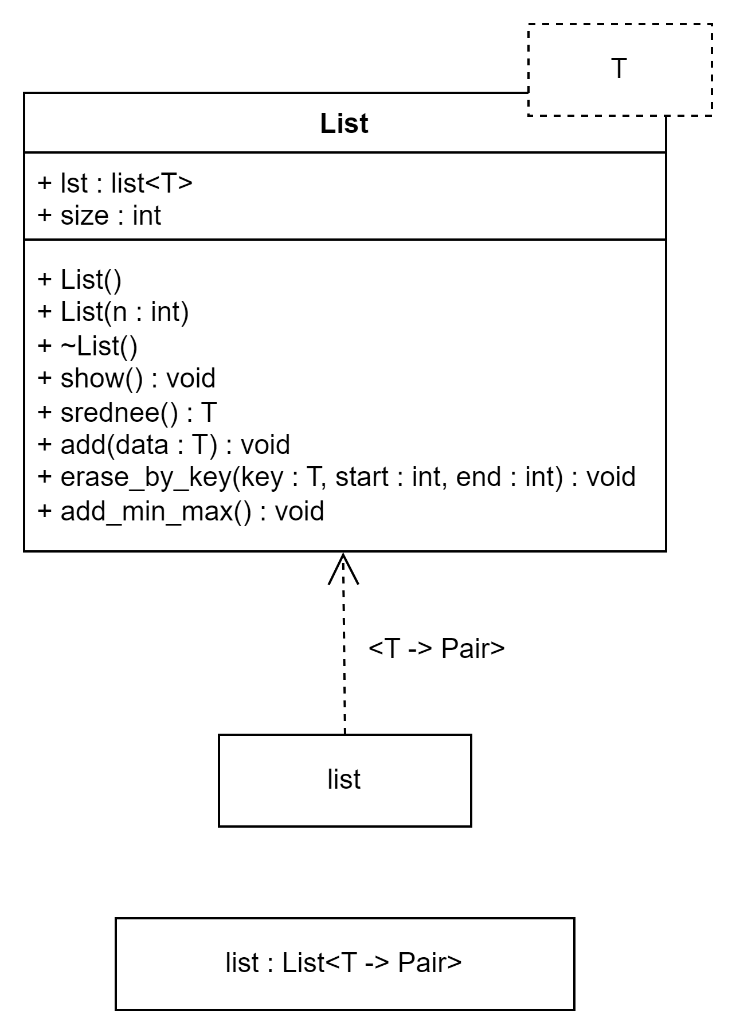


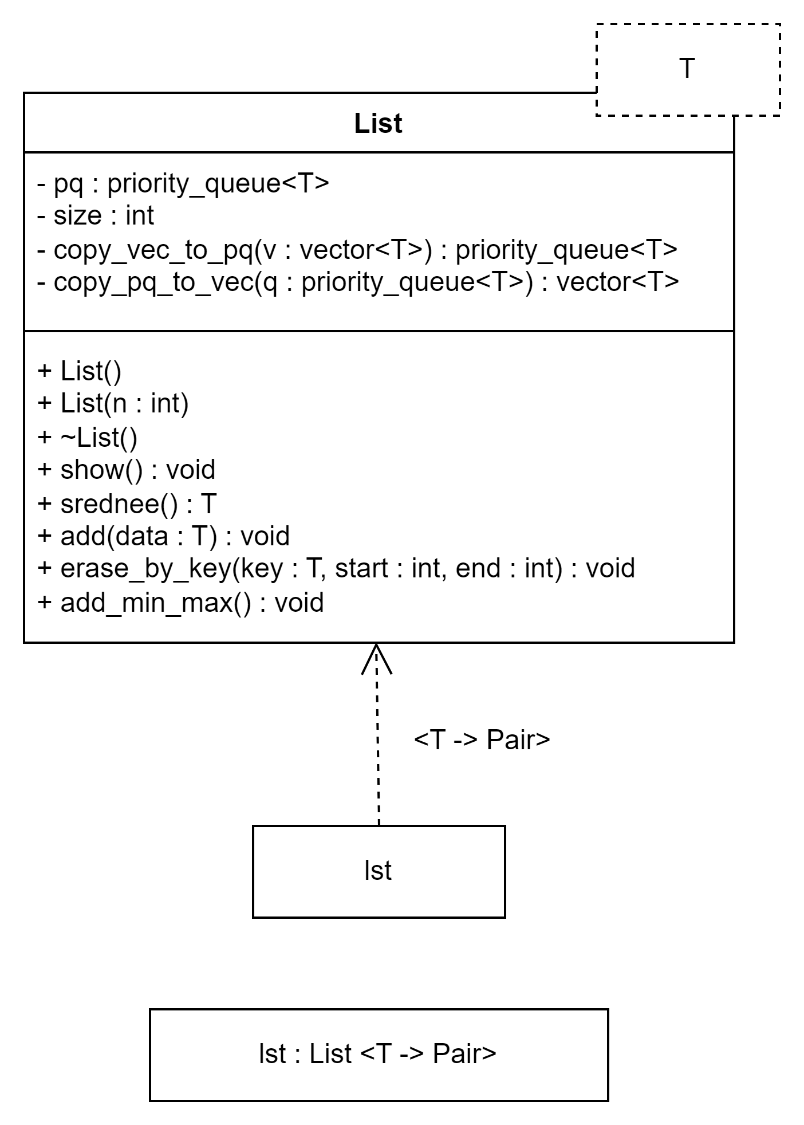




**UML-диаграммы**







**Ответы на контрольные вопросы**



STL – Standard Template Library, стандартная библиотека шаблонов состоит из двух основных частей: набора контейнерных классов и набора обобщенных алгоритмов.



Типы контейнеров:

1. Последовательные
   1. Основные
      1. Вектор
      2. Двунаправленная очередь
      3. Список
   2. Адаптеры
      1. Стек
      2. Очередь
      3. Очередь с приоритетами
2. Ассоциативные
   1. Словари
   2. Множества
   3. Словари с дубликатами
   4. Множества с дубликатами



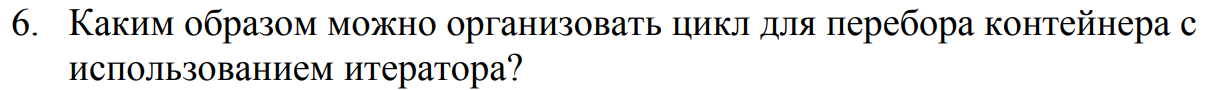
Для использования контейнера в программе необходимо подключить соответствующий заголовочный файл.

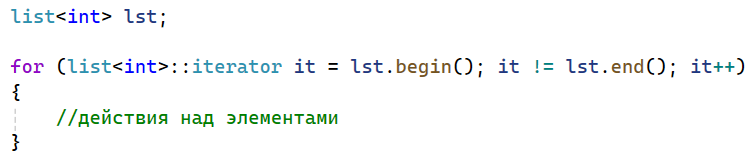


Указатели – обобщение концепции указателей. Они ссылаются на элементы контейнера и позволяют работу с ними.



* Разыменовывание
* Сравнение на равенство и неравенство
* Присваивание одного итератора другому
* Инкрементирование



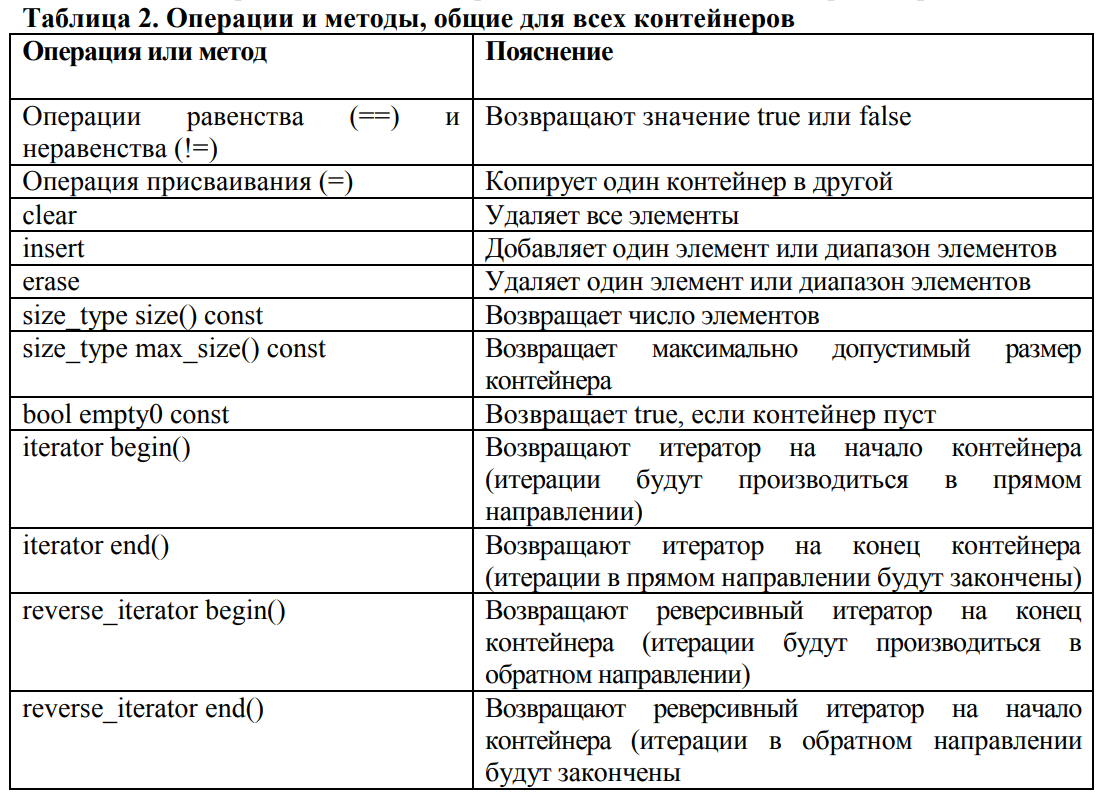




Типы итераторов:

* Входные
* Выходные
* Прямые
* Двунаправленные
* Произвольного доступа







Вектор эффективно обрабатывает элементы с помощью операции индексации [].

Неэффективными для вектора являются вставка и удаление элементов в любое место кроме конца, так как придется сдвигать элементы.

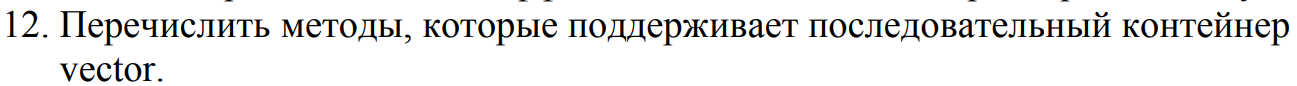


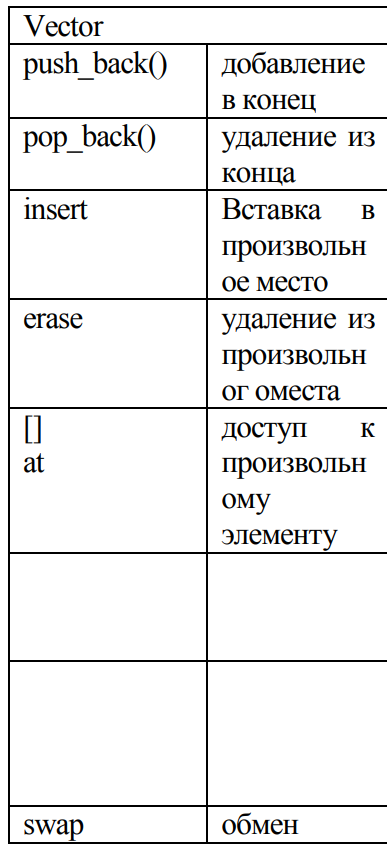
Для списка эффективны вставка и удаление элементов в любом месте списка, так как он основан на двусвязном списке и элементы связаны друг с другом при помощи указателей.

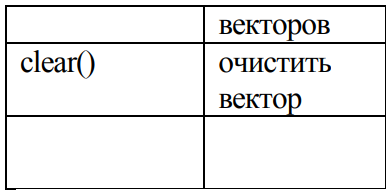
Но список не поддерживает произвольного доступа к элементам.



Двусторонняя очередь в целом похожа на вектор, но она также эффективна в вставке и удалении первого элемента.

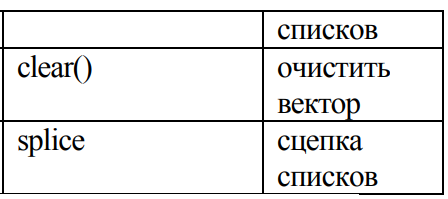


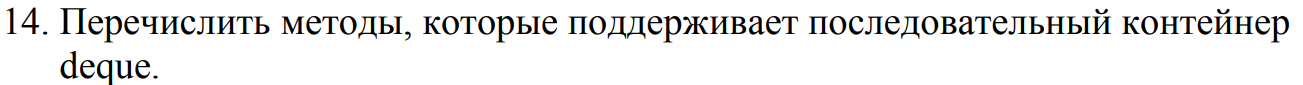






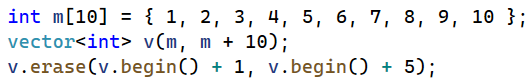




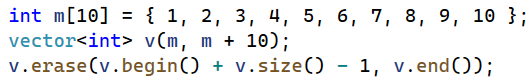




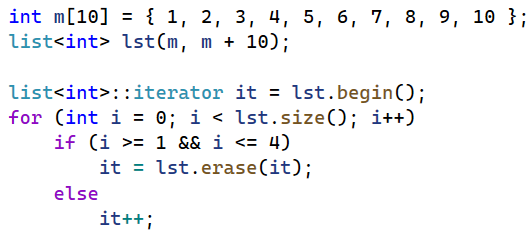




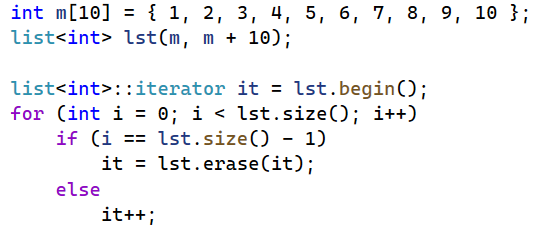




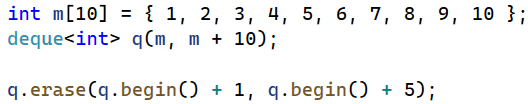




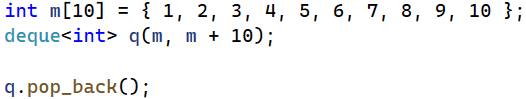


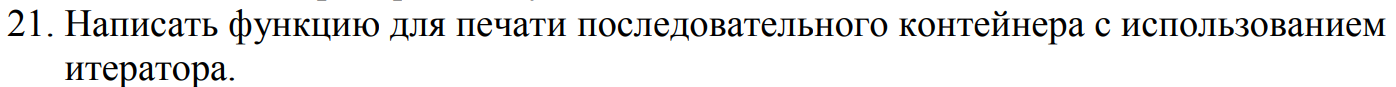


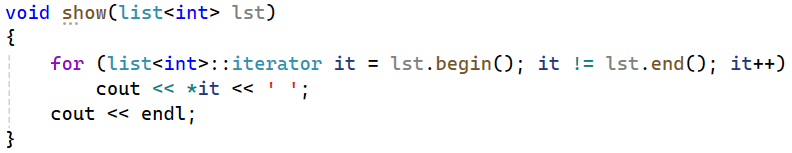














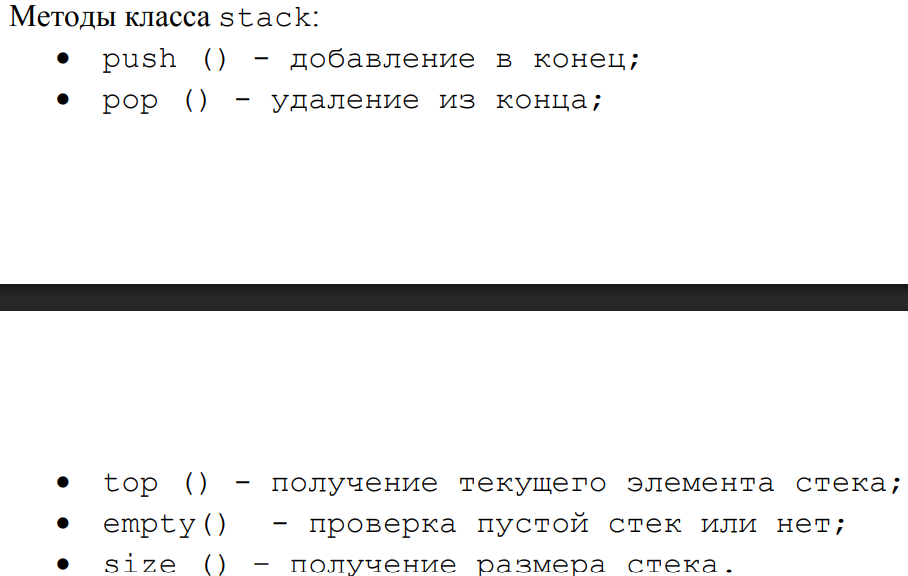
Адаптеры контейнеров – специализированные последовательные контейнеры, реализованные на базе основных последовательных контейнеров.



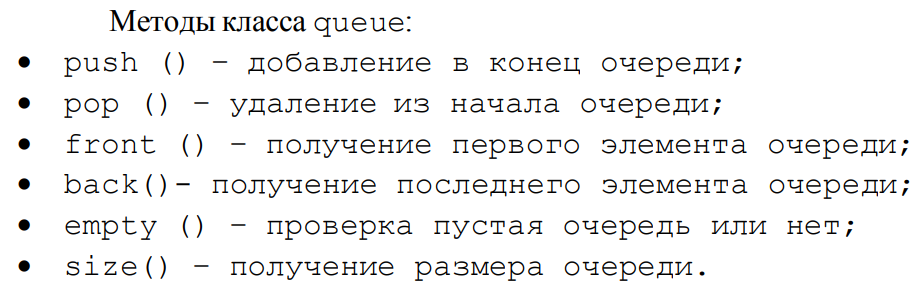
Объявление stack<int> создает стек на основе двусторонней очереди (по умолчанию)

Объявление stack<int, list<int>> создает стек на базе списка.





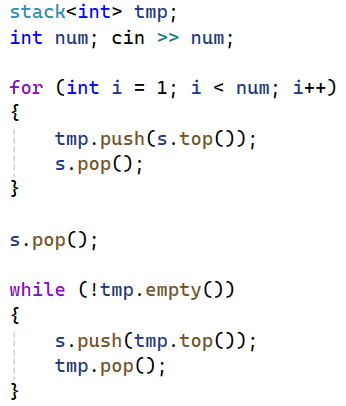




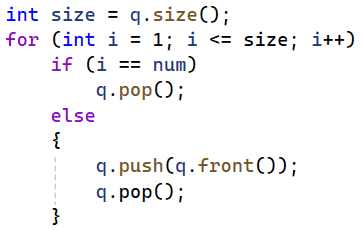


Очередь с приоритетами отличается от обычной очереди тем, что элементы из нее извлекаются по убыванию.

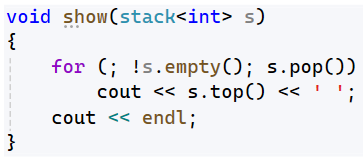




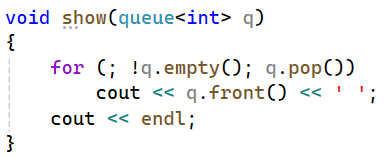












Ссылка на github: <https://github.com/RuntovIvan/Informatika>