Министерство образования и науки Российской Федерации

Федеральное государственное образовательное автономное учреждение высшего образования

"Пермский национальный исследовательский политехнический университет"

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Дисциплина: Основы алгоритмизации и программирования

Тема: АТД. Контейнеры

Вариант 15

|  |  |
| --- | --- |
| Выполнил работу | |
| Студент группы РИС-22-1б | |
| Мерзляков В.Н. | |
|  | |
| Проверил работу | |
| Доцент кафедры ИТАС | |
| Полякова О.А. | |
|  | |

Пермь – 2023

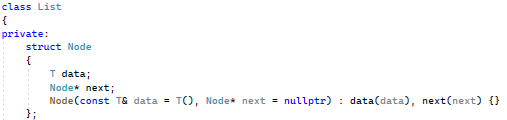
**Постановка задачи**

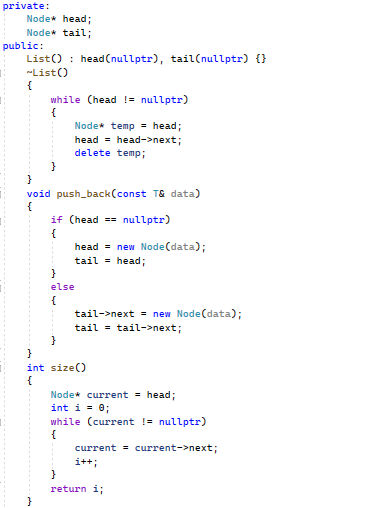
1. Определить класс-контейнер.
2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.
3. Перегрузить операции, указанные в варианте.
4. Реализовать класс-итератор. Реализовать с его помощью операции последовательного доступа.
5. Написать тестирующую программу, иллюстрирующую выполнение операций.

Вариант 15:

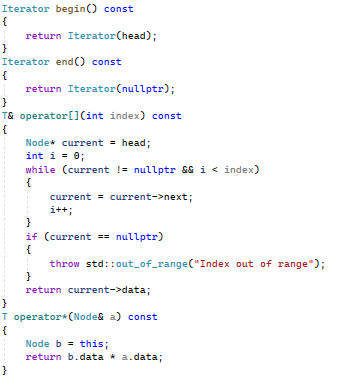
* Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции:
* [] – доступа по индексу;
* int() – определение размера списка;
* вектор – умножение элементов списков a[i]\*b[i];
* +n - переход вправо к элементу с номером n (с помощью класса-итератора).

**Описание классов**

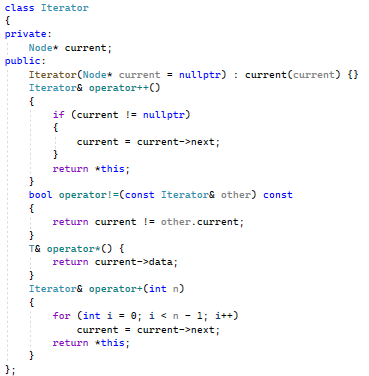
****

****

*Приложение 1 – Описание классов*

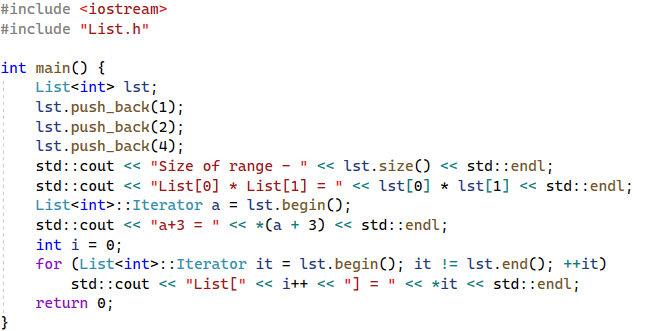
****

*Приложение 2 – Описание классов*

****

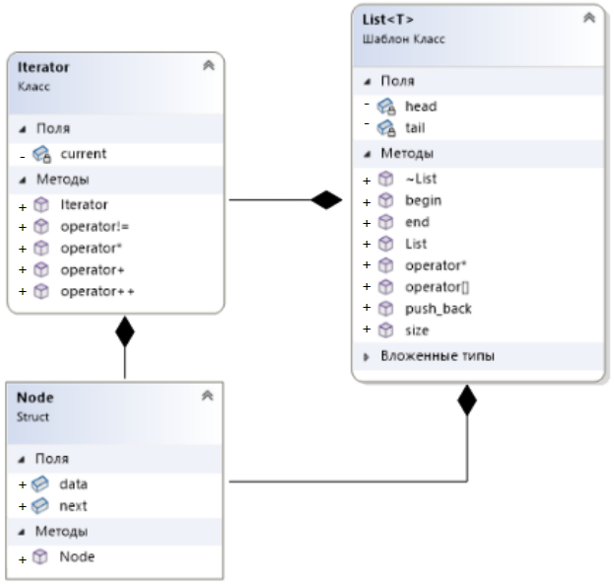
*Приложение 3 – Описание классов*

**Функция main()**

****

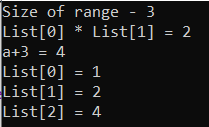
*Приложение 4 – Главная функция*

**UML-Диаграмма**



*Приложение 5 – UML-Диаграмма*

**Работа программы**



*Приложение 6 – Работа программы*

**Ответы на контрольные вопросы**

1. Что такое абстрактный тип данных? Привести примеры АТД.

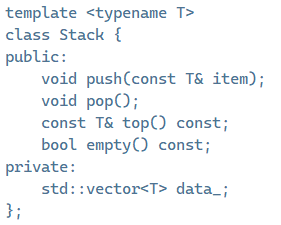
Абстрактный тип данных (АТД) - это тип данных, определенный не через его реализацию, а через набор операций, которые могут выполняться над этим типом данных. Операции определяются интерфейсом АТД и скрывают детали его реализации.

Примеры абстрактных типов данных:

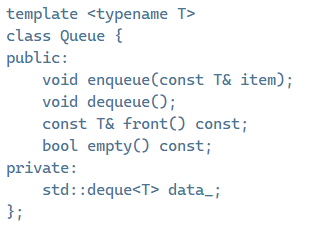
* Стек - это АТД, который поддерживает операции добавления и удаления элементов только в одном конце. Примеры реализации: стек на основе массива, стек на основе связного списка.
* Очередь - это АТД, который поддерживает операции добавления и удаления элементов в двух концах. Примеры реализации: очередь на основе массива, очередь на основе связного списка.
* Список - это АТД, который представляет собой набор элементов, упорядоченных в определенной последовательности. Он поддерживает операции добавления, удаления и поиска элементов. Примеры реализации: список на основе массива, список на основе связного списка.

1. Привести примеры абстракции через параметризацию.

Шаблонный класс "Стек" может быть параметризован типом элементов, которые он хранит.

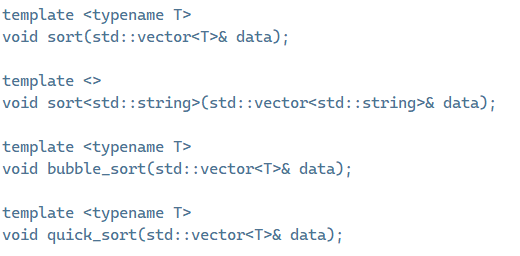


Шаблонный класс "Очередь" может быть параметризован типом элементов, которые он хранит.



1. Привести примеры абстракции через спецификацию.

Функция "сортировка" может быть специфицирована для различных типов данных и алгоритмов сортировки.



1. Что такое контейнер? Привести примеры.

Контейнер - это объект, который содержит и управляет другими объектами (элементами), обеспечивая доступ к ним и их хранение.

* std::vector - динамический массив, который может изменять свой размер во время выполнения программы.
* std::list - двусвязный список, который позволяет быстро добавлять и удалять элементы в любом месте списка.
* std::map - ассоциативный массив, который хранит пары ключ-значение и обеспечивает быстрый доступ к значению по ключу.
* std::set - упорядоченное множество уникальных элементов, которые могут быть быстро добавлены или удалены из множества.
* std::queue - очередь, которая работает по принципу "первым пришел - первым обслужен".
* std::stack - стек, который работает по принципу "последним пришел - первым обслужен".

1. Какие группы операций выделяют в контейнерах?

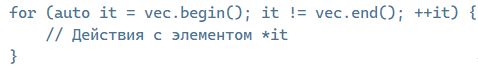
В контейнерах выделяют три группы операций:

* Операции добавления и удаления элементов - включают в себя методы pushback, pushfront, insert, erase и другие.
* Операции доступа к элементам - включают в себя методы operator, at, front, back и другие.
* Операции изменения размера контейнера - включают в себя методы resize, reserve и другие.

1. Какие виды доступа к элементам контейнера существуют? Привести примеры.

Существуют следующие виды доступа к элементам контейнера:

* Доступ по индексу - осуществляется с помощью оператора или метода at. Например, для вектора vec доступ к элементу с индексом i можно получить следующим образом: veci или vec.at(i).
* Доступ к первому и последнему элементам - осуществляется с помощью методов front и back соответственно. Например, для списка lst доступ к первому и последнему элементу можно получить следующим образом: lst.front() и lst.back().
* Итераторный доступ - осуществляется с помощью итераторов, которые позволяют перебирать элементы контейнера по порядку. Например, для вектора vec итераторный доступ можно реализовать следующим образом:



1. Что такое итератор?

Итератор - это объект, который используется для перебора элементов контейнера. Он представляет собой указатель на текущий элемент и может быть инкрементирован или декрементирован, чтобы переместиться к следующему или предыдущему элементу. Итераторы могут быть константными или неконстантными, в зависимости от того, можно ли изменять элементы контейнера через итератор или нет. Итераторы позволяют работать с контейнером без необходимости знать его внутреннюю структуру и реализацию.

1. Каким образом может быть реализован итератор?

Итератор может быть реализован как отдельный класс или структура, который содержит указатель на текущий элемент контейнера и методы для перемещения по контейнеру. Методы могут включать инкрементирование, декрементирование, получение значения текущего элемента, проверку на равенство и неравенство итераторов и т.д. Также может быть реализовано несколько различных типов итераторов для разных целей, таких как итераторы только для чтения, итераторы для изменения элементов и т.д.

1. Каким образом можно организовать объединение контейнеров?

Существует несколько способов объединения контейнеров, в зависимости от требуемой функциональности и типа контейнеров:

* Использование алгоритмов стандартной библиотеки. Стандартная библиотека C++ предоставляет множество алгоритмов для работы с контейнерами, включая алгоритмы для объединения контейнеров. Например, функция std::merge() объединяет два отсортированных контейнера в один отсортированный.
* Использование специальных контейнеров. Некоторые контейнеры, такие как std::set и std::map, предоставляют методы для объединения с другими контейнерами. Например, метод std::set::insert() может быть использован для объединения двух множеств.
* Ручное объединение. Если необходимо объединить контейнеры с определенной логикой, можно написать свою функцию или метод для объединения. Например, для объединения двух списков можно создать новый список и добавлять в него элементы из обоих списков поочередно.
* Использование шаблонов. Если требуется объединить контейнеры разных типов, можно использовать шаблоны. Например, можно написать шаблонную функцию, которая принимает два контейнера и возвращает новый контейнер, содержащий элементы обоих контейнеров.

1. Какой доступ к элементам предоставляет контейнер, состоящий из элементов «ключ-значение»?

Контейнер, состоящий из элементов "ключ-значение", предоставляет доступ к элементам по ключу. Например, в контейнере std::map можно получить значение элемента по его ключу с помощью метода operator. Также доступ к элементам можно получить с помощью итераторов, которые указывают на пары "ключ-значение".

1. Как называется контейнер, в котором вставка и удаление элементов выполняется на одном конце контейнера?

Такой контейнер называется стек (stack).

1. Какой из объектов является контейнером?

* int mas=10;
* int mas;
* struct {char name[30]; int age;} mas;
* int mas[100];

Контейнером является int mas[100]; - это массив, который может содержать до 100 элементов типа int.

1. Какой из объектов не является контейнером?

* int a[]={1,2,3,4,5};
* int mas[30];
* struct {char name[30]; int age;} mas[30];
* int mas;

Объект int mas; не является контейнером, так как это просто переменная типа int, которая может хранить только одно значение.

1. Контейнер реализован как динамический массив, в нем определена операция доступ по индексу. Каким будет доступ к элементам контейнера?

Доступ к элементам контейнера осуществляется по индексу, используя оператор или метод at(). Например, для доступа к элементу с индексом i в контейнере container необходимо использовать выражение containeri или container.at(i).

1. Контейнер реализован как линейный список. Каким будет доступ к элементам контейнера?

Доступ к элементам контейнера, реализованного как линейный список, осуществляется по индексу через итераторы. Для доступа к элементу с индексом i необходимо пройти по списку от начала до i-го элемента, используя итераторы. Таким образом, сложность доступа к элементу в линейном списке равна O(n), где n - количество элементов в списке.