ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа №6

АТД. Контейнеры

Вариант №15

Выполнил студент группы РИС-23-3Б

Блинов А. Е.

Проверила доцент кафедры ИТАС

О. А. Полякова

2024 г.

**Постановка задачи**

1. Определить класс-контейнер.

2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.

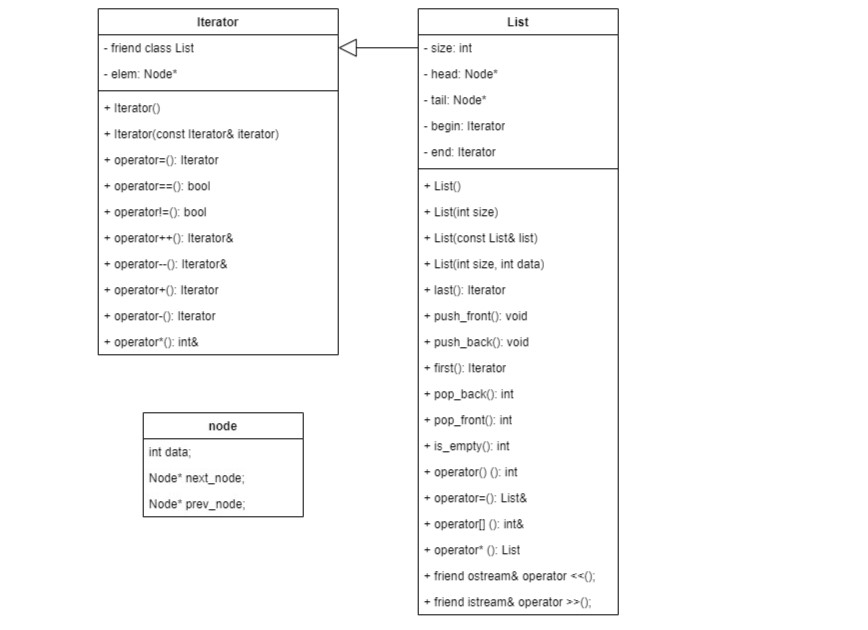
3. Перегрузить операции, указанные в варианте.

4. Реализовать класс-итератор. Реализовать с его помощью операции последовательного доступа.

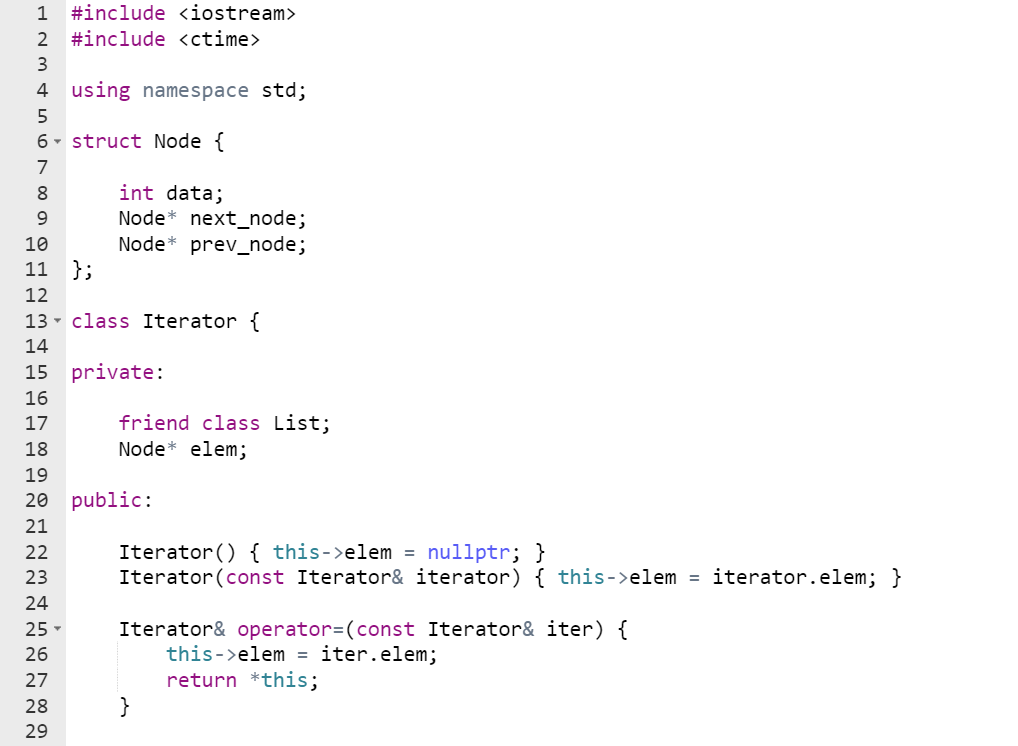
5. Написать тестирующую программу, иллюстрирующую выполнение операций.

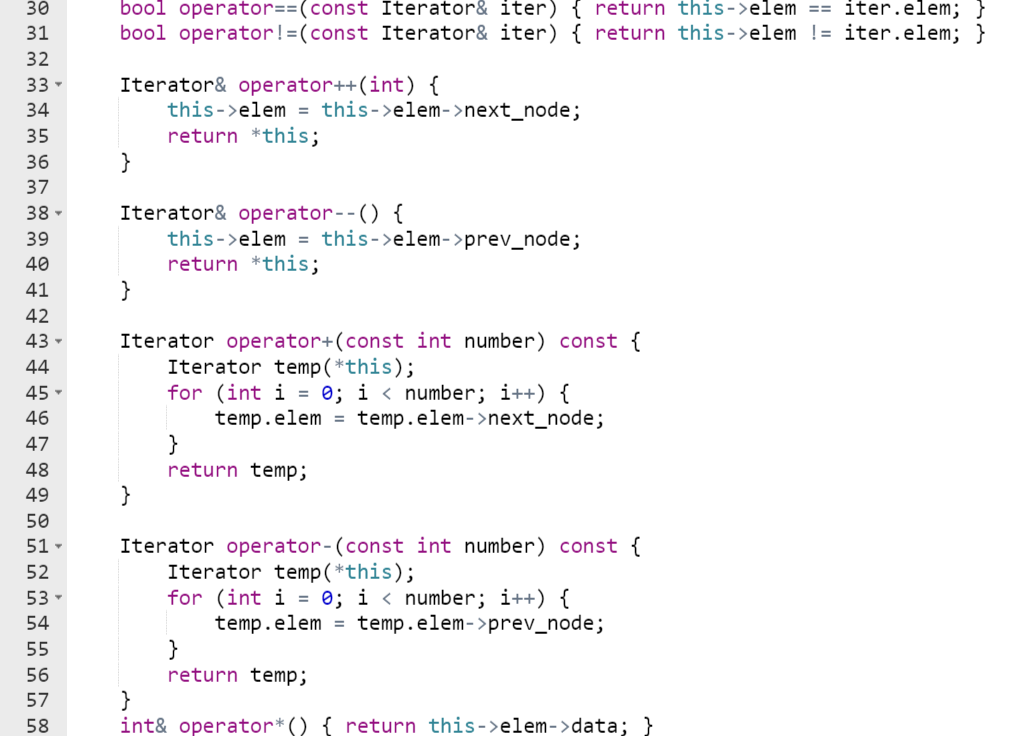
Класс- контейнер СПИСОК с ключевыми значениями типа int. Реализовать операции: [] – доступа по индексу; int() – определение размера списка; \* вектор – умножение элементов списков a[i]\*b[i]; +n - переход вправо к элементу с номером n ( с помощью класса-итератора).

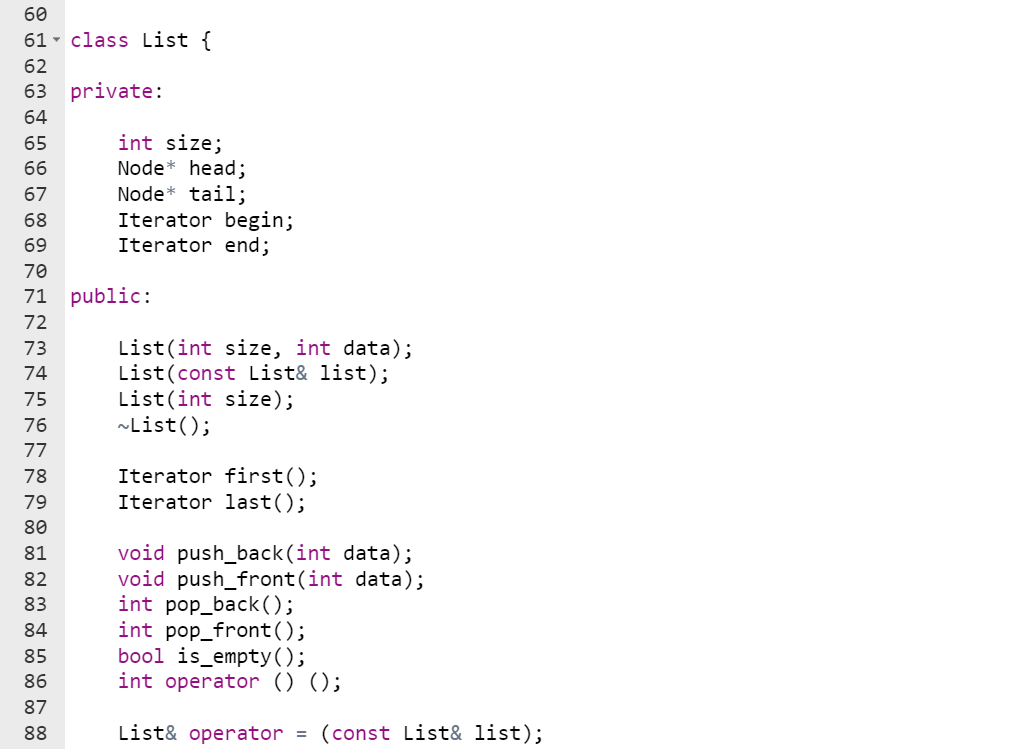
**UML диаграмма**

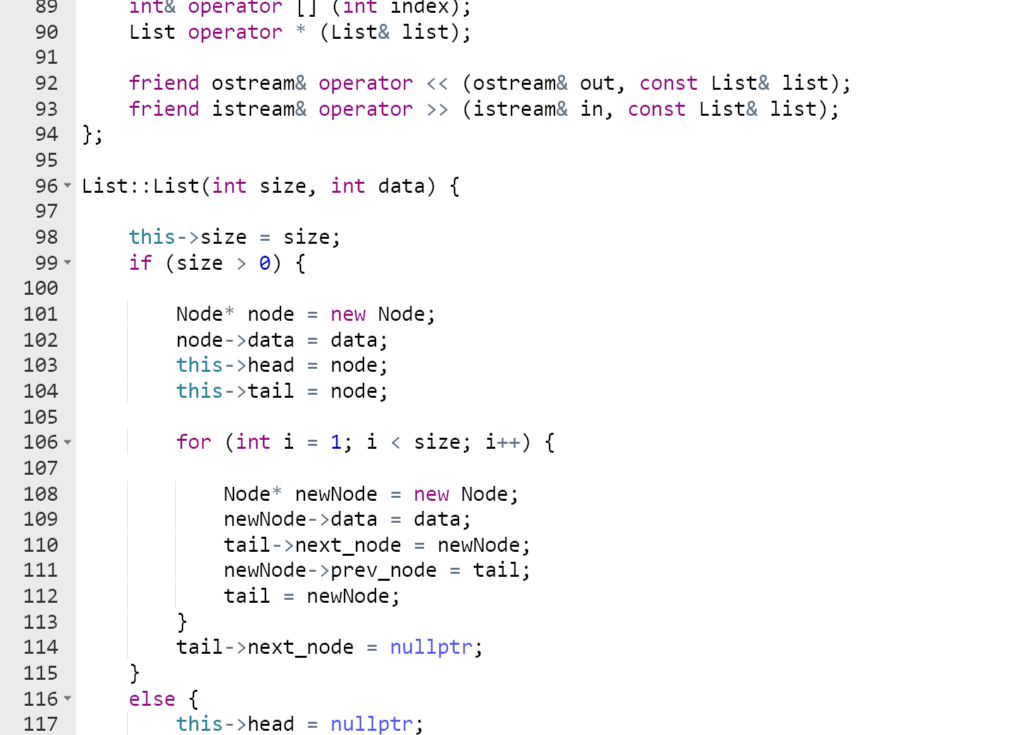


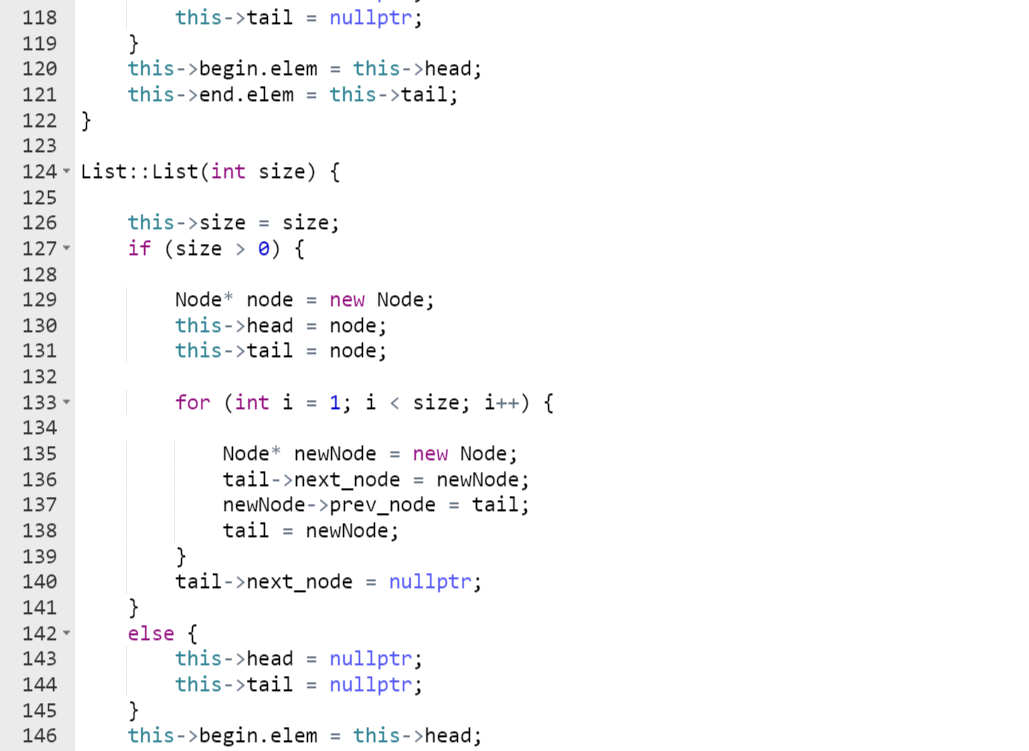
**Код на языке C++**

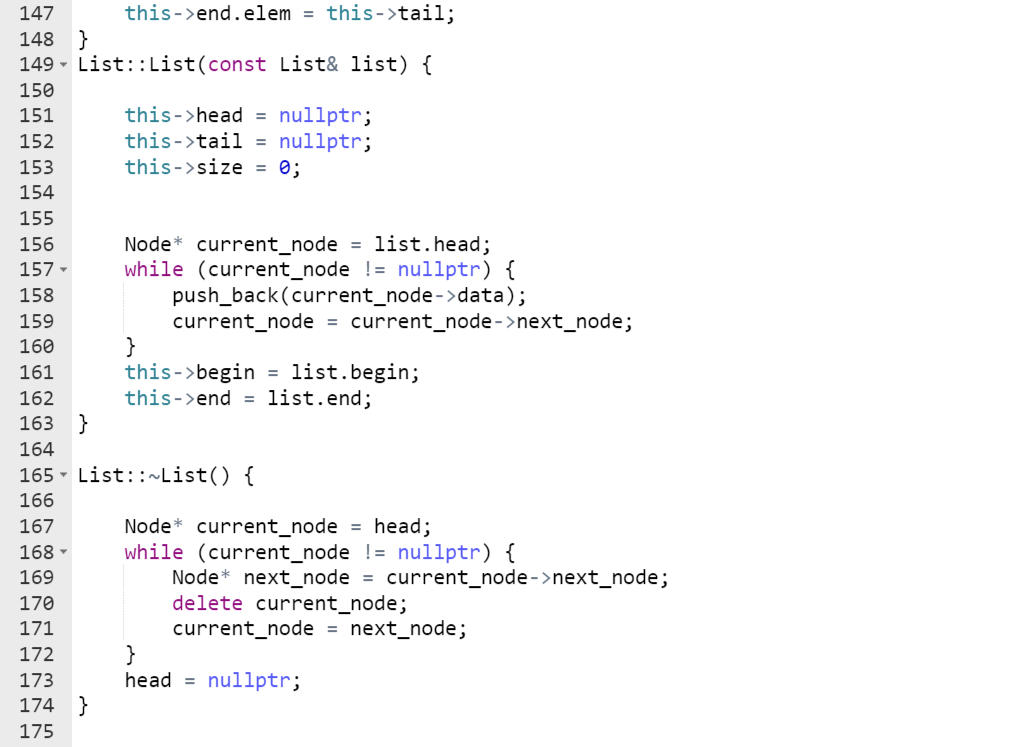
****

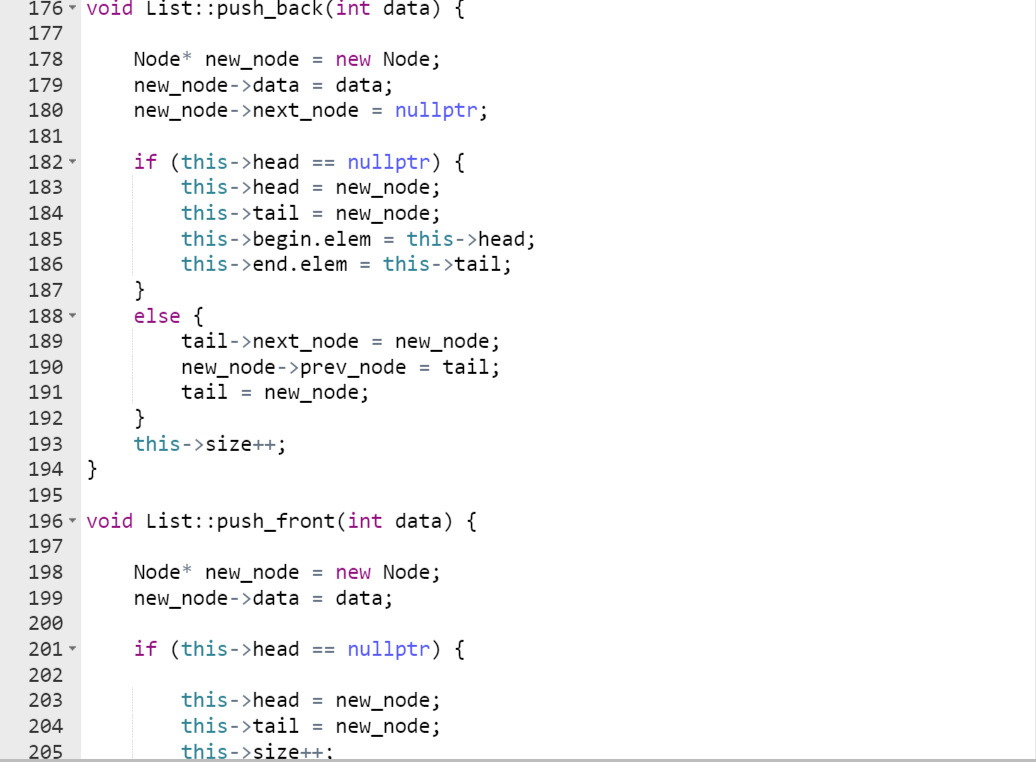
****

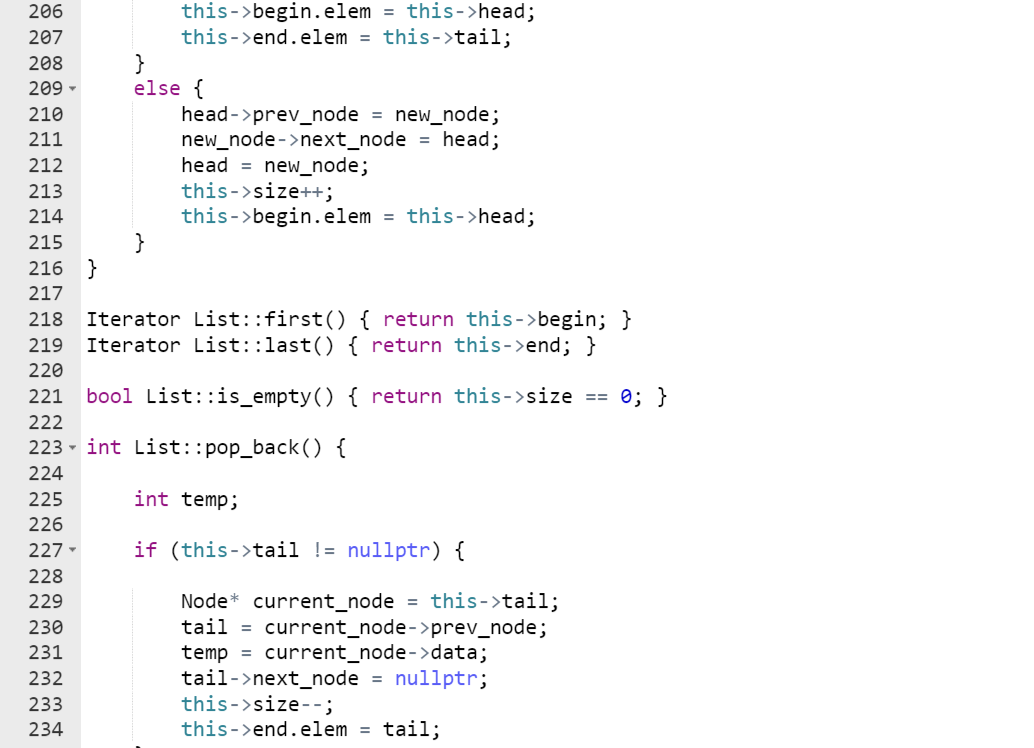
****

****

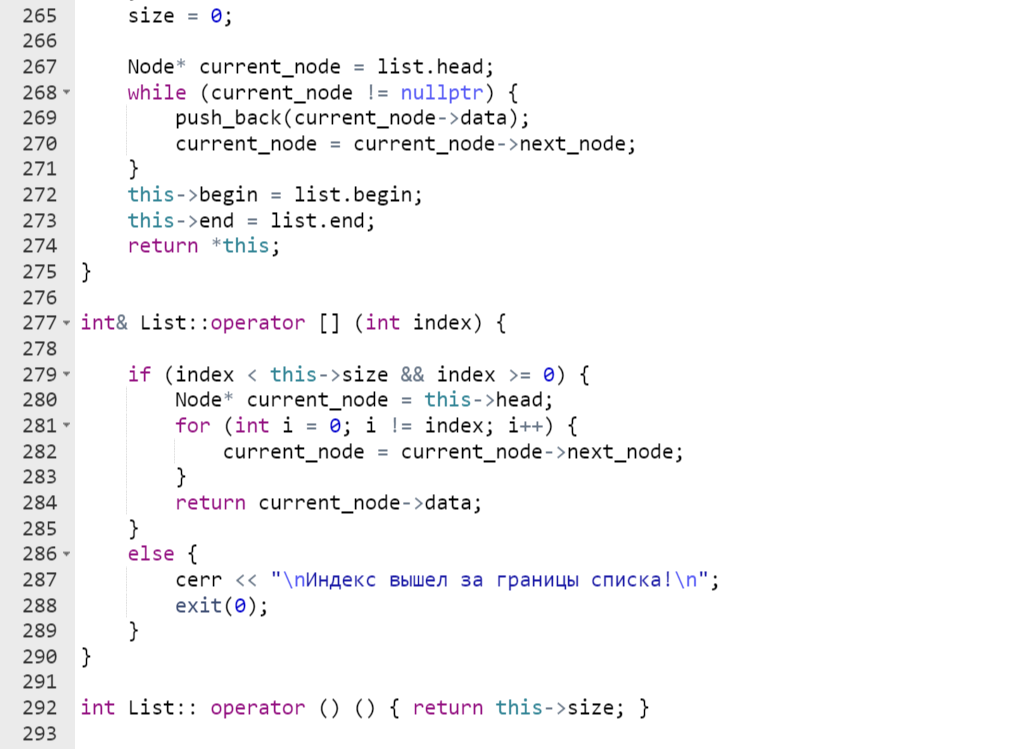
****

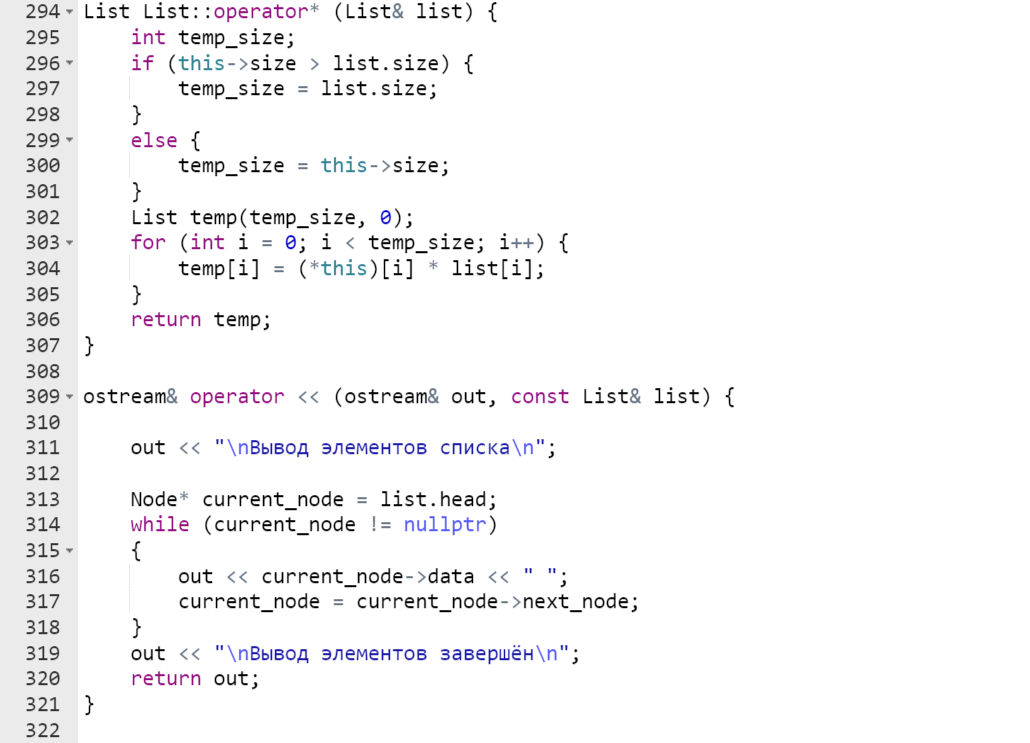
****

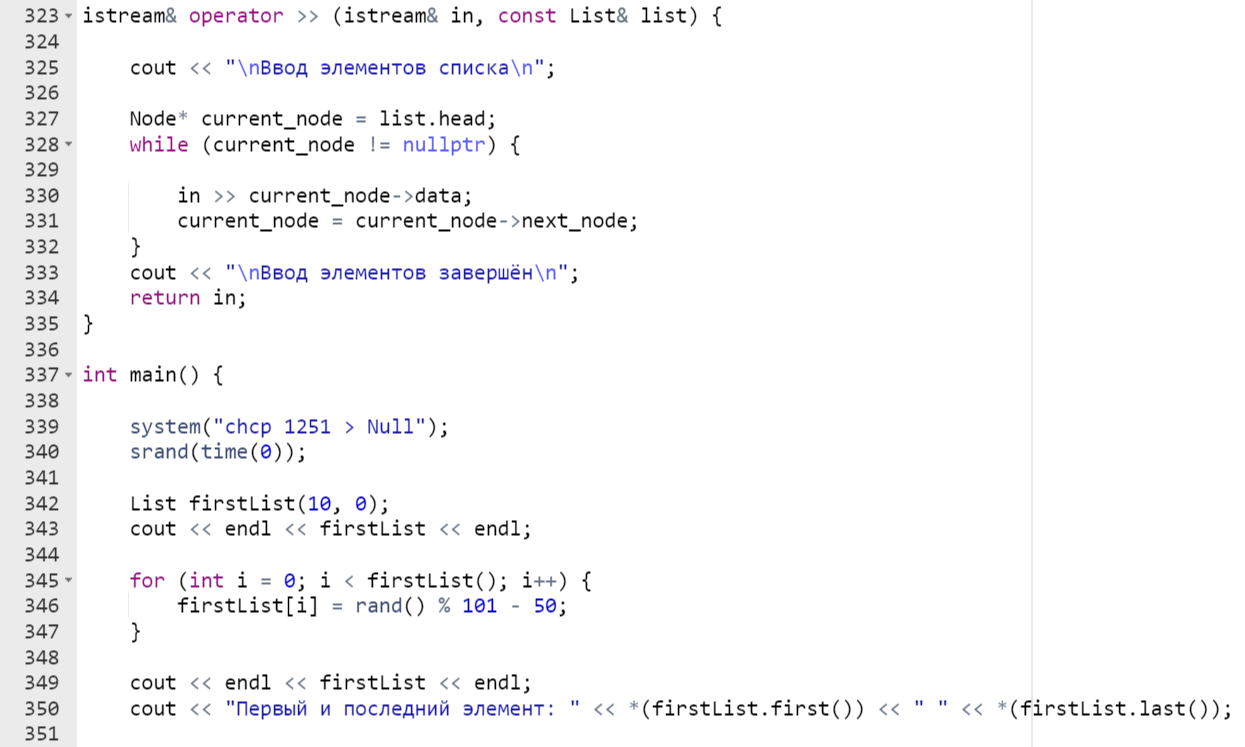
****

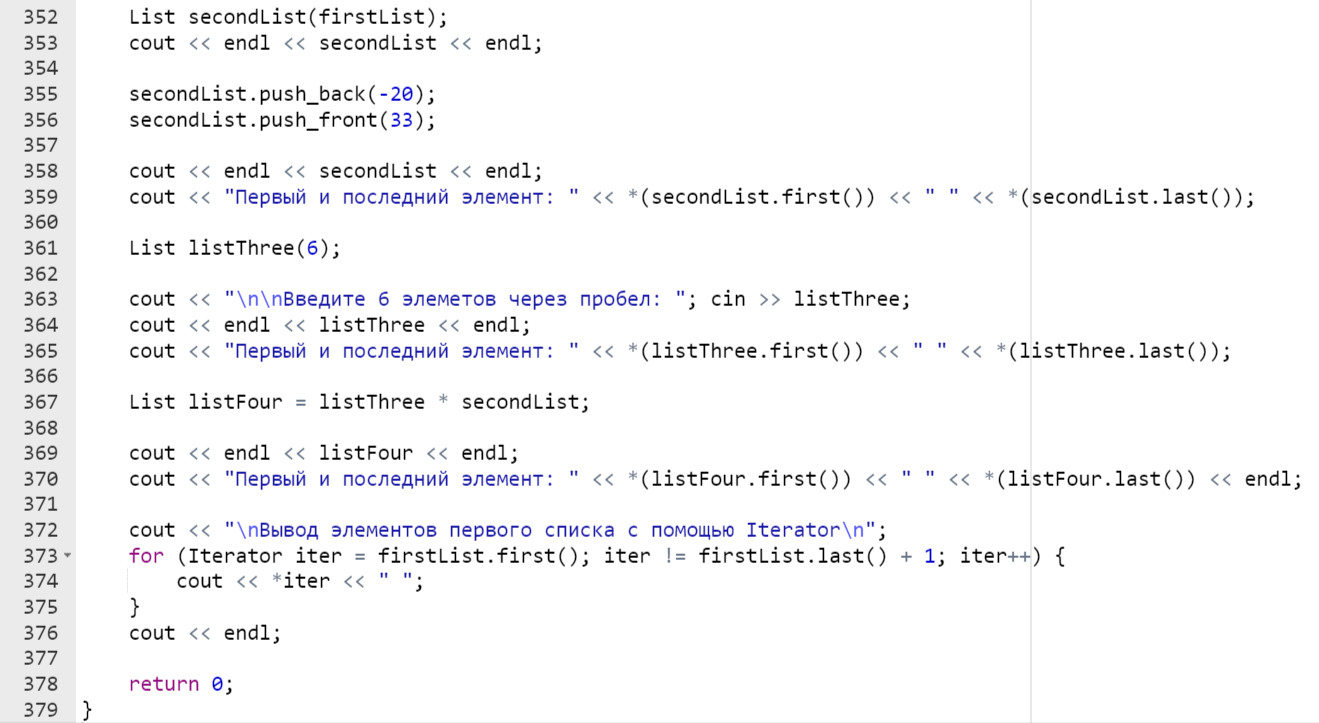
****

****

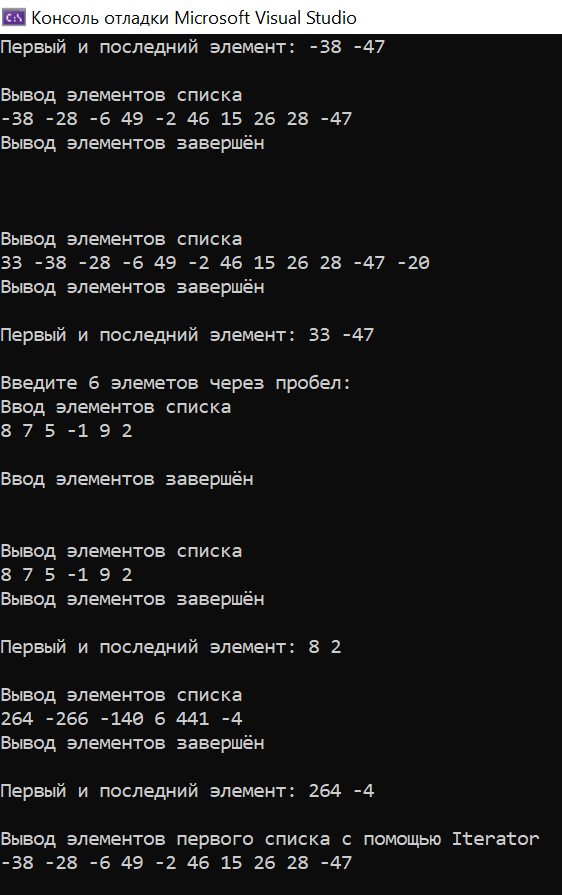
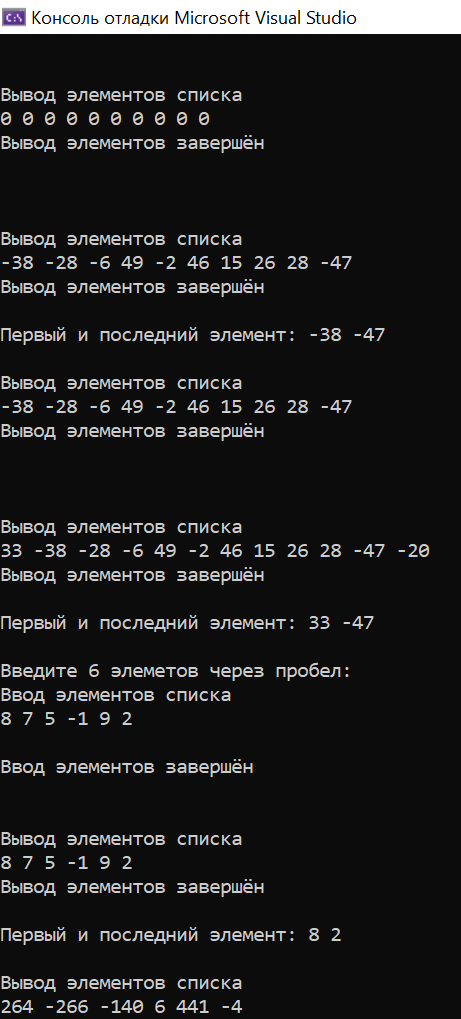
****

****

****

****

**Результаты работы программы**

****

**Выводы**

Программа работает успешно

GitHub: https://github.com/alexblin123/Laba-6-OOP.git

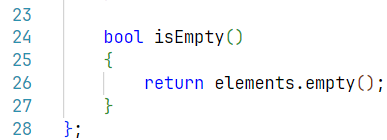
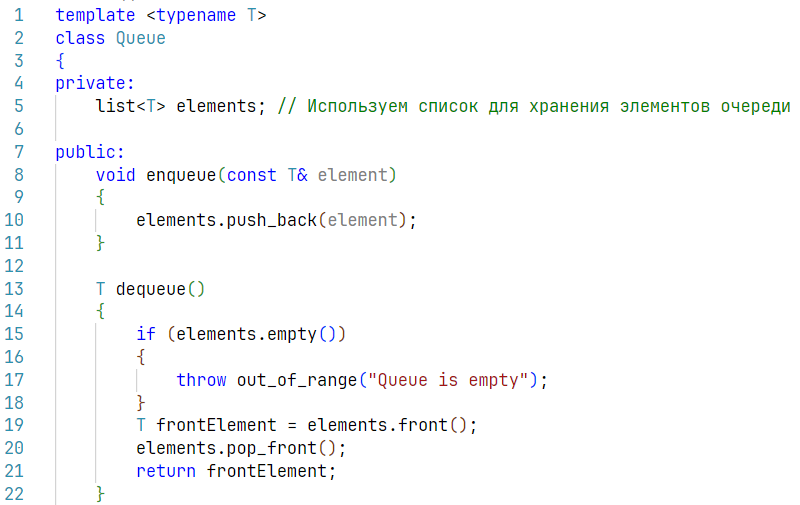
**Контрольные вопросы**

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

Абстрактный тип данных (АТД) — это модель для типов данных, где тип определяется его поведением, изложенным в виде списка операций, доступных с данными типа, без каких-либо указаний на то, как операции должны быть реализованы. (примеры: очередь, двунаправленный список)

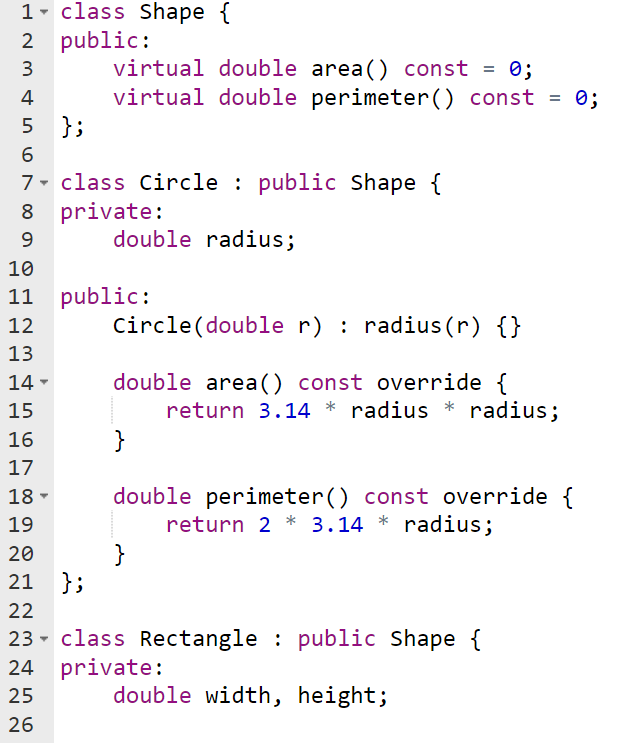
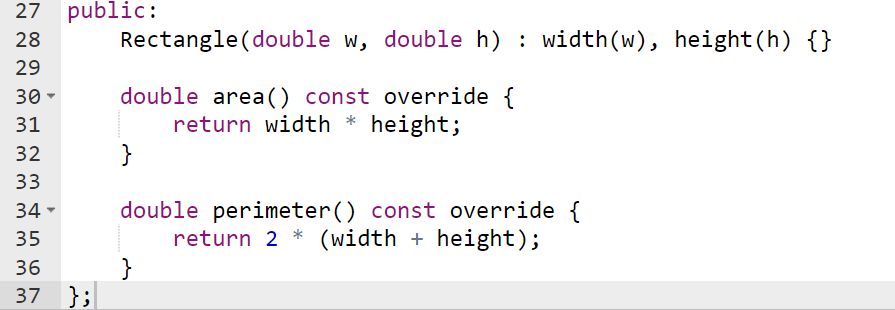
**2. Привести примеры абстракции через параметризацию.**

Абстракция через параметризацию (АП) — это подход, позволяющий создавать гибкие и масштабируемые абстракции, используя параметры для определения общих структур или алгоритмов без привязки к конкретным типам данных.



**3. Привести примеры абстракции через спецификацию.**

**Абстракция через спецификацию (АС)** - позволяет абстрагироваться от процесса вычислений, до уровня знания того, что данная функция должна в итоге реализовать. Это достигается путем задания для каждой функции спецификации, описывающей эффект ее работы. При этом смысл обращения к функции становится ясным через анализ ее спецификации, а не тела процедуры. (пример ниже)

**4. Что такое контейнер? Привести примеры.**

**Контейнер** – набор однотипных элементов. (например: массивы, двусвязные списки)

**5. Какие группы операций выделяют в контейнерах?**

* Операции доступа к элементам, которые обеспечивают и операцию замены значений элементов;
* Операции добавления и удаления элементов или групп элементов;
* Операции поиска элементов и групп элементов;
* Операции объединения контейнеров;
* Специальные операции, которые зависят от вида контейнера.

**6. Какие виды доступа к элементам контейнера существуют? Привести примеры.**

Существуют различные виды доступа к элементам контейнера в C++: доступ по индексу, доступ через итераторы. Например, для вектора доступ к элементам можно получить как myVector[index], для списка - с помощью итераторов std::list::iterator, для карты (map) - через ключ myMap[key].

**7. Что такое итератор?**

**Итератор** — абстрактный тип данных, предоставляющий доступ к элементам коллекции (массива или контейнера) и навигацию по ним.

**8. Каким образом может быть реализован итератор?**

Итератор может быть реализован различными способами: в виде указателя или даже в виде функции-члена контейнера.

**9. Каким образом можно организовать объединение контейнеров?**

Для объединения контейнеров в C++ можно использовать различные подходы в зависимости от типов контейнеров.

Например, если у вас есть два вектора, и вы хотите объединить их в один, вы можете использовать функцию std::merge из стандартной библиотеки, которая объединяет отсортированные диапазоны в один новый вектор. Также можно просто скопировать элементы одного контейнера в другой с помощью цикла или использовать специализированные алгоритмы, предоставляемые STL для объединения контейнеров определенных типов.

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

Контейнер, состоящий из элементов "ключ-значение" в C++, представляет собой ассоциативный контейнер, позволяющий хранить пары значений, где каждый элемент состоит из ключа и соответствующего ему значения. (например, хэш-таблица)

Доступ к элементам в таких контейнерах осуществляется по ключу, который обеспечивает быстрый поиск значений.

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

Очередь

**12. Какой из объектов (a,b,c,d) является контейнером?**

a. int mas=10;

b. int mas;

c. struct {char name[30]; int age;} mas;

d. int mas[100];

Ответ: объект d. Так как объект d представляет собой массив int mas[100].

**13. Какой из объектов (a,b,c,d) не является контейнером?**

a. int a[]={1,2,3,4,5};

b. int mas[30];

c. struct {char name[30]; int age;} mas[30];

d. int mas;

Ответ: объект d. Так как объект d является простой переменной типа int.

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

Для доступа к элементам контейнера, реализованного как динамический массив с возможностью доступа по индексу, используется операция индексации [].

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

Для доступа к элементам контейнера, реализованного как линейный список, обычно используется итерация по списку с помощью указателей или итераторов.