Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЁТ**

**по дисциплине «Информатика»**

Семестр: 2

На тему: « Абстрактные типы данных. Контейнеры.»

Выполнил студент ИВТ-22-2б:

Зайцев Павел Денисович

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

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.

**Варивант 9:**

Класс- контейнер МНОЖЕСТВО с элементами типа int.

Реализовать операции:

* []-достапа по индексу;
* !=-проверки на неравенство;
* < число- принадлежность числа множеству;
* +n -переход вправо к элементу с номером n.

**Описание класса**

Класс plenty:

Поля типа private: size-переменная целочисленного типа, data-укзатель на динамический массив значений множества, beg-указатель на первый элемент множества, end-указатель на элемент следующий за последним.

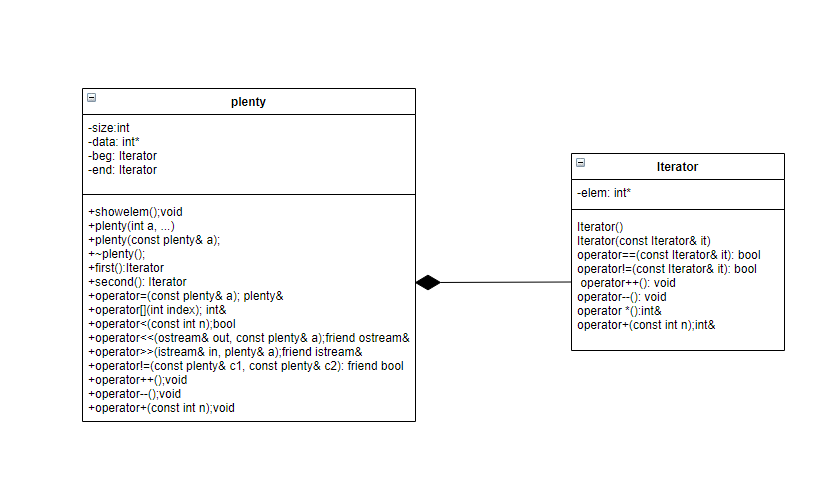
Методы типа public:2 вида конструкторов (с параметрами, копирования), деструктор, перегруженный оператор сравнения(!=), перегруженный оператор разыменования(\*), перегруженные операторы потокового ввода-вывода, перегруженный оператор принадлежности числа множеству(<n), перегруженный оператор доступа по индексу([]).

Класс Iterator:

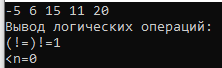
Поле типа private: elem: int\*.

Методы типа public: 2 вида конструкторов (с параметрами, копирования), деструктор, перегруженные операторы сравнения(==, !=), перегруженные операторы инкремента и декремента(++, --), перегруженный оператор разыменования(\*), перегруженный оператор перевода итератора вправо.

Также описание классов представлено на UML-диаграмме.



**Вывод программы**



**Контрольные вопросы**

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

Абстрактный тип данных - это набор, включающий данные и выполняемые над ними операции. Примеры: список, стек, очередь.

2. Привести примеры абстракции через параметризацию.

template<class ss>

void print(ss m) {

cout << m;

}

3. Привести примеры абстракции через спецификацию.

class Vector

{

public:

int& operator[](int index)

{

if (index < size) return data[index];

else cout << "\nError! Index>size";

}

private:

int size;

int\* data;

}

4. Что такое контейнер? Привести примеры.

Контейнер – это объект, содержащий другие однотипные объекты. Примеры: двусвязный список, массив, стек, множество.

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.

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

Виды доступа: последовательный, прямой и ассоциативный.

Последовательный доступ – перемещение от элемента к элементу. Пример: a.next.

Прямой доступ – это доступ по индексу. Например, a[10].

Ассоциативный доступ также выполняется по индексу, но индексом будет являться не номер элемента, а его содержимое. Пример: a[“word”].

7. Что такое итератор?

Итератор – это объект, который обеспечивает последовательный доступ к элементам контейнера.

8. Каким образом может быть реализован итератор?

Итератор реализуется как класс, который имеет такой же интерфейс, как и указатель для совместимости с массивами.

9. Каким образом можно организовать объединение контейнеров?

Наиболее часто используется операция объединения двух контейнеров с получением нового контейнера. Она может быть реализована в разных вариантах:

* Простое сцепление двух контейнеров: в новый контейнер попадают сначала элементы первого контейнера, потом второго, операция не коммутативна.
* Объединение упорядоченных контейнеров, новый контейнер тоже будет упорядочен, операция коммутативна.
* Объединение контейнеров как объединение множеств, в новый контейнер попадают только те элементы, которые есть хотя бы в одном контейнере, операция коммутативна.
* Объединение контейнеров как пересечение множеств, в новый контейнер попадают только те элементы, которые есть в обоих контейнерах, операция коммутативна.

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

Ассоциативный доступ.

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

Стек.

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

a. int mas=10;

b. 2. int mas;

c. 3. struct {char name[30]; int age;} mas;

d. 4. int mas[100];

Ответ: d

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

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

b. 2. int mas[30];

c. 3. struct {char name[30]; int age;} mas[30];

d. 4. int mas;

Ответ: d

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

Прямой доступ.

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

Последовательный доступ.