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: «Шаблоны классов»

Выполнил студент ИВТ-22-2б:

Зайцев Павел Денисович

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

**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.

**Вариант 9:**

Класс- контейнер МНОЖЕСТВО с элементами типа ии.

Реализовать операции:

[]доступа по индексу;

!=- проверка на неравенство;

< n число — принадлежность числа множеств:

Пользовательский класс Money для работы с денежными суммами. Число должно

‘быть представлено двумя полями: типа long для рублей и типа int для копеек.

Дробная часть числа при выводе на экран должна быть отделена от целой части.

запятой.

**Описание классов**

Класс plenty

Поля типа private: size - целочисленного типа, data - указатель шаблонного типа данных.

Методы типа public: 2 вида конструкторов и деструктор; перегрузки операторов присваивания operator=(const Money&), доступа по индексу operator[](int index), оператор принадлежности константы ко всем элементам вектора operator<(const T k),перегруженный оператор неравенства(!=).

Класс Money.

Поля типа private: long rub – первое число длинное целого типа,

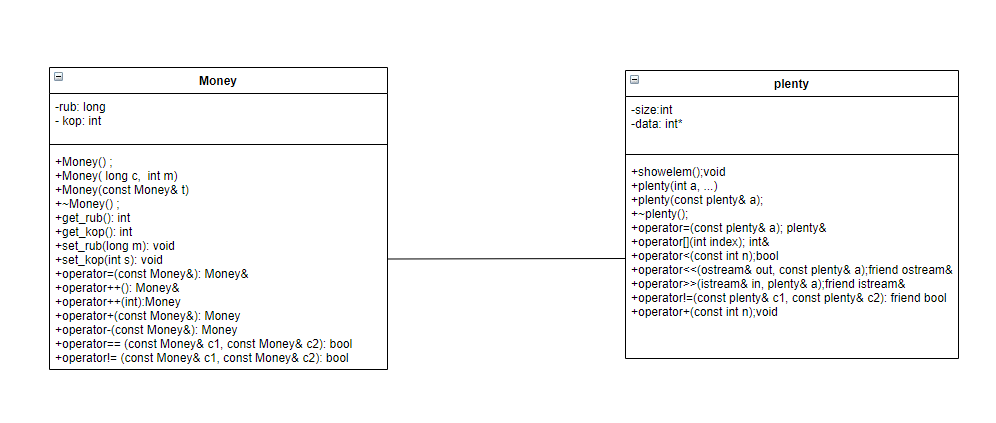
Int b – второе число целого типа.

Методы типа public: 3 вида конструкторов и деструктор; геттеры и сеттеры

для полей; перегрузки операторов присваивания operator=(const Money&), вычитания экземпляров класса operator-(const Money&), сложения для экземпляров класса operator+(const Money&),

сложения для дробных чисел operator+(double), перегруженный операторы сравнения экземпляров класса operator== (const Money& c1, const Money& c2), operator!= (const Money& c1, const Money& c2).

Также описание класса представлено на UML-диаграмме.



**Описание функций**

Необходимо реализовать перегрузку операций ввода/вывода в виде дружественных функций:

friend ostream& operator<< (ostream& out, const Money&);

friend istream& operator>> (istream& in, Money&);

**Результат работы 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.