Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

Тема: «Лабораторная работа №12»

Выполнил

Студент группы РИС-22-2б

Прядеин И.А.

Проверил доцент кафедры

ИТАС

Полякова О. А.

Пермь 2023

Постановка задачи

Задача 1.

1. Создать ассоциативный 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 ассоциативный 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 параметризованный класс, используя в качестве контейнера ассоциативный 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 параметризованного класса.

**Вариант 8:**

**Задача 1.**

1. Контейнер - map.
2. Тип элементов - double.

**Задача 2.**

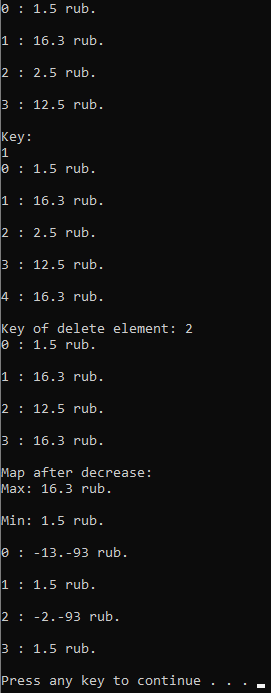
Тип элементов Money

**Задача 3.**

Параметризованный класс - Вектор

|  |  |  |
| --- | --- | --- |
| **Задание 3** | **Задание 4** | **Задание 5** |
| Найти элемент с заданным ключом и добавить его на заданную позицию контейнера | Найти элемент с заданным ключом и удалить его из контейнера | Найти разницу между максимальным и минимальным элементами контейнера и вычесть её из каждого элемента контейнера |

**Скриншот результата выполнения программы**



**Ответы на вопросы:**

1. Что представляет собой ассоциативный контейнер?

**Ответ:** Ассоциативный контейнер - это обобщение понятия ассоциативного массива, содержащего пары значений: **ключ** (key) - **отоборажаемое значение** (mapped value).

1. Перечислить ассоциативные контейнеры библиотеки STL.

**Ответ:** Ассоциативные контейнеры библотеки STL:

1. Словарь (map)
2. Словарь дубликатами (multimap)
3. Множество (set)
4. Множество дубликатами (multiset)
5. Каким образом можно получить доступ к элементам ассоциативного контейнера?

**Ответ:** Для доступа к элементам ассоциативного контейнера используется итератор. В отличие от последовательных контейнеров, в ассоциативных контейнерах элементы хранятся не в порядке их добавления, а в отсортированном порядке на основе ключа. Поэтому для доступа к элементам по индексу используют итераторы.

1. Привести примеры методов, используемых в ассоциативных контейнерах.

**Ответ:** Основные методы, используемые в ассоциативных контейнерах:

* bool empty() const, size() const, max\_size() - Методы определения размеров
* insert() - Добавление элемента или диапазона элементов
* erase() - Удаление одного элемента или диапазона элементов
* clear() - Удаление всех элементов
* swap() - Обмен данными с контейнером того же типа
* find(), count(), lower\_bound(), upper\_bound() - Методы поиска

1. Каким образом можно создать контейнер map? Привести примеры.

**Ответ:** map<string, int> m;

1. Каким образом упорядочены элементы в контейнере map по умолчанию? Как изменить порядок на обратный?

**Ответ:** Элементы в контейнере map упорядочены в порядке возрастания ключей. Для изменения порядка на обратный можно использовать следующий вариант реализации создания контейнера.

bool compare(int el1, int el2) {

return el1 > el2;

}

map<int, string, decltype(compare)\*> m(compare);

1. Какие операции определены для контейнера map?

**Ответ:** Для контейнера map определены следующие операции:

* Добавление, удаление, поиск и доступ по ключу элемента
* Проверка наличия элементов в контейнере

1. Написать функцию для добавления элементов в контейнере map с помощью функции make\_pair().

**Ответ:**

template <class Tf, class Ts>

void insert\_elements(map<Tf, Ts>& m, int numberOfEl) {

Tf temp1 = 0;

Ts temp2;

for (int i = 0; i < n; i++) {

cin >> temp2;

m.insert(make\_pair(temp1++, temp));

}

}

1. Написать функцию для добавления элементов в контейнер map с помощью функции операции прямого доступа [].

**Ответ:**

template <class T1, class T2>

void insert\_elements(map<T1, T2>& m, T1 key, T2 value) {

m[key] = value;

}

1. Написать функцию для печати контейнера map с помощью итератора.

**Ответ:**

template <class T1, class T2>

void print\_map(map<T1, T2>& m) {

for (auto it = m.begin(); it != m.end(); it++)

cout << it->first << “ : “ << it->second << “\n”;

}

1. Написать функцию для печати контейнера map с помощью функции операции прямого доступа [].

**Ответ:**

template <class T1, class T2>

void print\_map(map<T1, T2>& m) {

for (const auto& p : m)

cout << p.first << “ : “ << m[p.first] << endl;

}

1. Чем отличаются контейнеры map и multimap?

**Ответ:** Словари с дубликатами (multimap), в отличие от словарей (map), допускают хранение элементов с одинаковыми ключами.

1. Что представляет собой контейнер set?

**Ответ:** Контейнер set - упорядоченное множество уникальных элементов.

1. Чем отличаются контейнеры map и set?

**Ответ:** Контейнер set, в отличие от контейнера map, хранит уникальные элементы без пары **ключ - значение**.

1. Каким образом можно создать контейнер set? Привести примеры.

**Ответ:** set<int> s;

vector<int> vec = {2, 5, 7, 1}

set<int> s(vec.begin(), vec.end());

1. Каким образом упорядочены элементы в контейнере set по умолчанию? Как изменить порядок на обратный?

**Ответ:** Элементы в контейнере set упорядочены по возрастанию.

1. Какие операции определены для контейнера set?

**Ответ:** Для контейнера set определены следующие методы:

- insert() - Добавление элемента в контейнер

- erase() - Удаление элемента из контейнера

- find() - Поиск элемента

- size() - Возвращает количество элементов в контейнере

- empty() - Возвращает true, если контейнер пуст, в противном случае false

- clear() - Удаляет все элементы контейнера

1. Написать функцию для добавления элементов в контейнер set.

**Ответ:**

addElements(set<int>& s, int n) {

for (int i = 0; i < n; i++)

s.insert(rand() % 11);

}

1. Написать функцию для печати контейнера set.

**Ответ:**

void print\_set(set<int>& s) {

for (const auto& i : s)

cout << i << endl;

}

1. Чем отличаются контейнеры set и multiset?

**Ответ:** Контейнеры set и multiset отличаются в хранении уникальных данных. В контейнере multiset есть возможность в хранении нескольких одинаковых элементов.