ПНИПУ

Пермский национальный исследовательский политехнический университет

Лабораторная работа ООП

Выполнил студент ИВТ-23-2б

Меновщиков Глеб Николаевич

Проверила:

Доцент кафедры ИТАС

Ольга Андреевна Полякова

2024

**11 Вариант**.

### Задача 1.

* + - 1. Создать последовательный контейнер.
      2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.
      3. Заменить элементы в соответствии с заданием (использовать алгоритмы

replace\_if(), replace\_copy(), replace\_copy\_if(), fill()).

* + - 1. Удалить элементы в соответствии с заданием (использовать алгоритмы

remove(),remove\_if(), remove\_copy\_if(),remove\_copy())

* + - 1. Отсортировать контейнер по убыванию и по возрастанию ключевого поля (использовать алгоритм sort()).
      2. Найти в контейнере заданный элемент (использовать алгоритмы find(), find\_if(), count(), count\_if()).
      3. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

* + - 1. Для выполнения всех заданий использовать стандартные алгоритмы библиотеки STL.

### Задача 2.

1. Создать адаптер контейнера.
2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.
3. Заменить элементы в соответствии с заданием (использовать алгоритмы

replace\_if(), replace\_copy(), replace\_copy\_if(), fill()).

1. Удалить элементы в соответствии с заданием (использовать алгоритмы

remove(),remove\_if(), remove\_copy\_if(),remove\_copy())

1. Отсортировать контейнер по убыванию и по возрастанию ключевого поля (использовать алгоритм sort()).
2. Найти в контейнере элемент с заданным ключевым полем (использовать

алгоритмы find(), find\_if(), count(), count\_if()).

1. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

1. Для выполнения всех заданий использовать стандартные алгоритмы библиотеки STL.

### Задача 3

1. Создать ассоциативный контейнер.
2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.
3. Заменить элементы в соответствии с заданием (использовать алгоритмы

replace\_if(), replace\_copy(), replace\_copy\_if(), fill()).

1. Удалить элементы в соответствии с заданием (использовать алгоритмы

remove(),remove\_if(), remove\_copy\_if(),remove\_copy())

1. Отсортировать контейнер по убыванию и по возрастанию ключевого поля (использовать алгоритм sort()).
2. Найти в контейнере элемент с заданным ключевым полем (использовать

алгоритмы find(), find\_if(), count(), count\_if()).

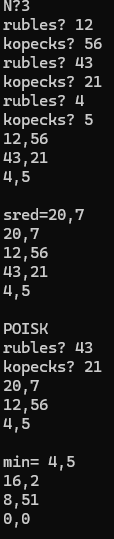
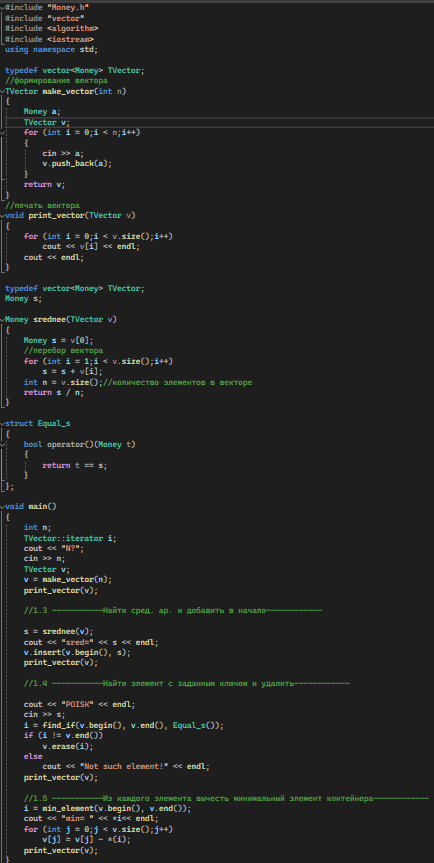
1. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

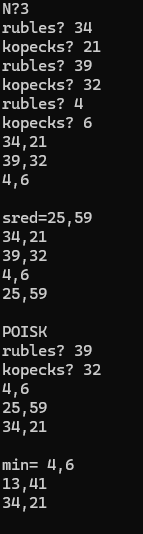
1. Для выполнения всех заданий использовать стандартные алгоритмы библиотеки STL.

|  |  |  |  |
| --- | --- | --- | --- |
| 11 | **Задача 1**   1. Контейнер - вектор 2. Тип элементов Money (см. лабораторную работу №3).   **Задача 2**  Адаптер контейнера - очередь.  **Задача 3**  Ассоциативный контейнер - словарь | | |
| **Задание 3** | **Задание 4** | **Задание 5** |
| Найти среднее арифметическое и добавить его в начало  контейнера | Найти элемент с заданным ключом и удалить их из контейнера | Из каждого элемента вычесть минимальный элемент контейнера |

1 Задача



2 Задача



3 Задача

