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: «Стандартные обобщенные алгоритмы библиотеки STL»

Выполнил студент ИВТ-22-2б:

Зайцев Павел Денисович

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

***Постановка задачи***

**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()).

4. Удалить элементы в соответствии с заданием (использовать алгоритмы

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

5. Отсортировать контейнер по убыванию и по возрастанию ключевого поля

(использовать алгоритм sort()).

6. Найти в контейнере заданный элемент (использовать алгоритмы find(),

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

7. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

8. 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()).

4. Удалить элементы в соответствии с заданием (использовать алгоритмы

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

5. Отсортировать контейнер по убыванию и по возрастанию ключевого поля

(использовать алгоритм sort()).

6. Найти в контейнере элемент с заданным ключевым полем (использовать

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

7. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

8. 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()).

4. Удалить элементы в соответствии с заданием (использовать алгоритмы

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

5. Отсортировать контейнер по убыванию и по возрастанию ключевого поля

(использовать алгоритм sort()).

6. Найти в контейнере элемент с заданным ключевым полем (использовать

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

7. Выполнить задание варианта для полученного контейнера (использовать алгоритм

for\_each()) .

8. Для выполнения всех заданий использовать стандартные алгоритмы библиотеки

STL.

Вариант №9

**Задача 1**

1. Контейнер - список

2. Тип элементов Pair (см. лабораторную работу №3).

**Задача 2**

Адаптер контейнера – очередь с приоритетами.

**Задача 3**

Ассоциативный контейнер - словарь

**Задание 3**

Найти среднее арифметическое и добавить его в конец контейнера

**Задание 4**

Найти элементы ключами из заданного диапазона и удалить их из контейнера

**Задание 5**

К каждому элементу добавить сумму минимального и максимального элементов контейнера.

**Описание класса**

Задача 1

1. Контейнер - deque

2. Тип элементов - Money

Задача 2

1. Параметризированный класс – Вектор (см. лабораторную работу №7)

2. Тип элементов - Money

Задача 3

1. Контейнер - set

2. Тип элементов - Money

**UML-диаграмма**

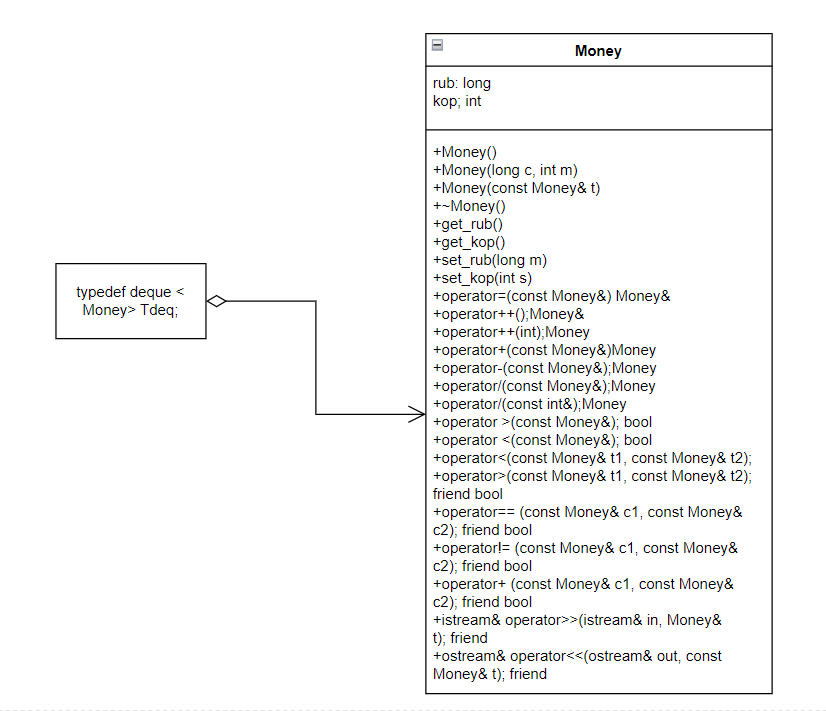


Диаграмма к задаче 1

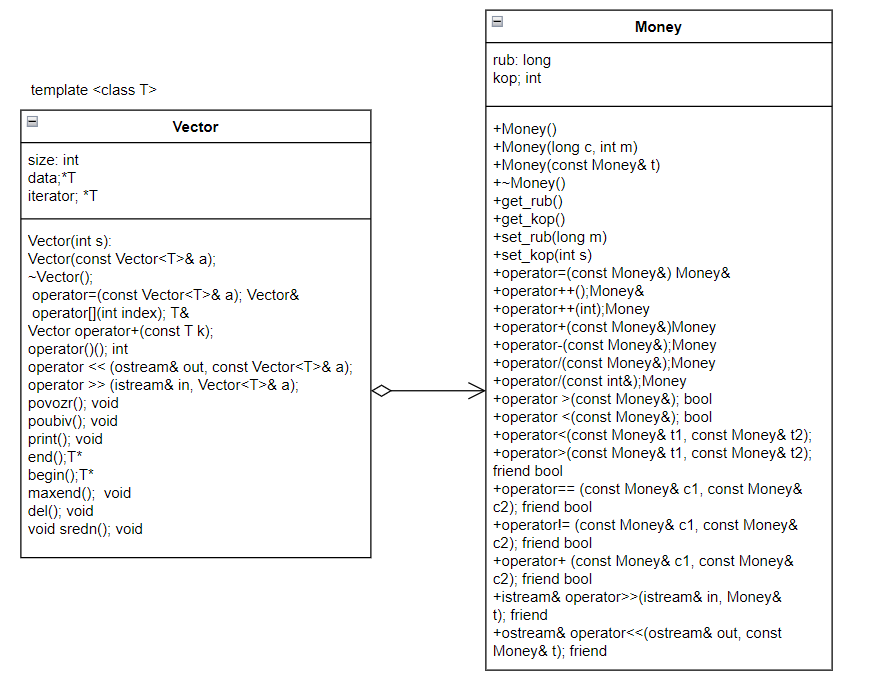


Диаграмма к задаче 2

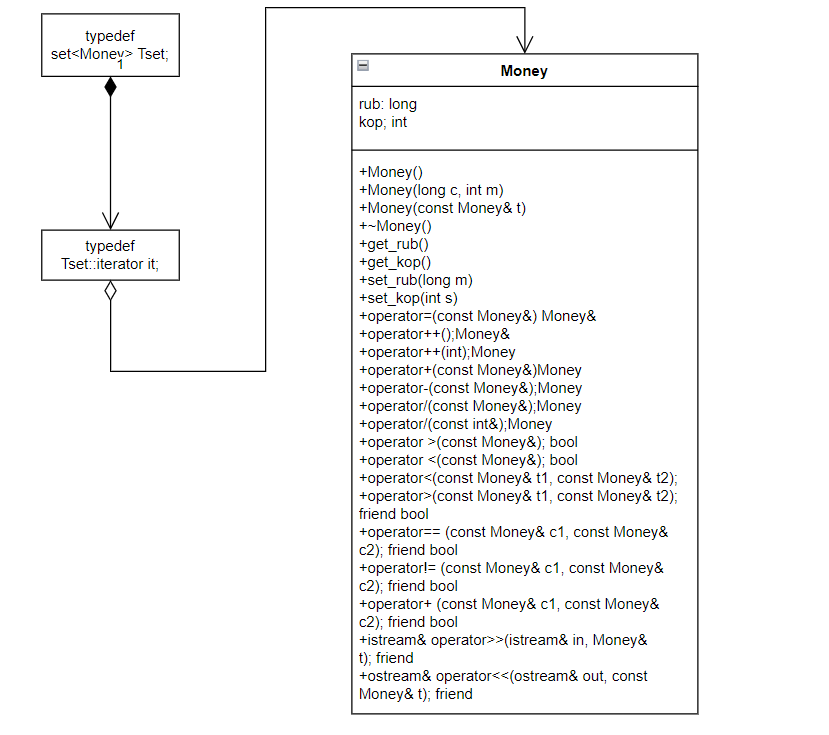


Диаграмма к задаче 3

**Скриншоты работы программы**

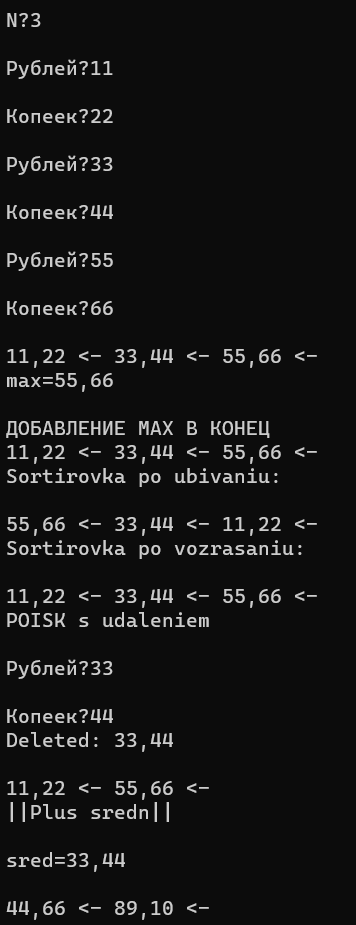


Рисунок 1 -Вывод программы из задачи 1

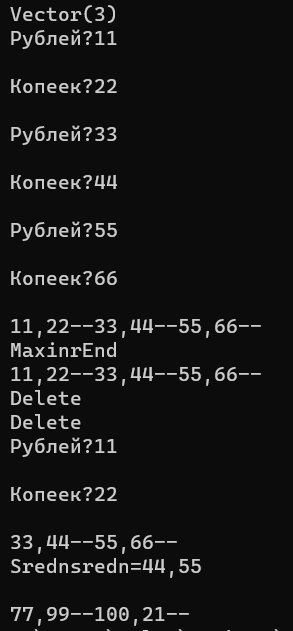


Рисунок 2 -Вывод программы из задачи 2

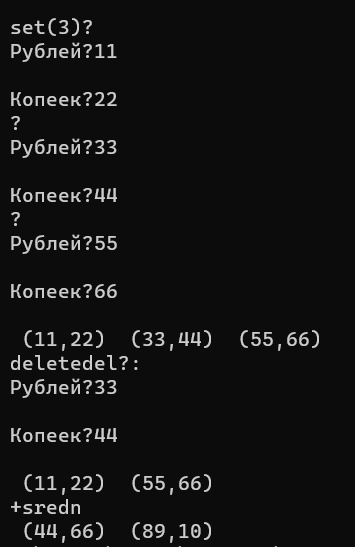


Рисунок 3 -Вывод программы из задачи 3