Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных технологий и анализа данных

АБСТРАКТНЫЕ СТРУКТУРЫ ДАННЫХ.

ОТЧЕТ  
по лабораторной работе №6  
Вариант №10

по дисциплине «Программирование»

Выполнил

Студент группы ЭВМб-23-3 \_\_\_\_\_\_\_\_\_\_\_\_Г. М. Распутин  
  
Принял

\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедева Е.Н.

Иркутск 2024

## Введение в библиотеки программирования

## Условие задачи.

**Общая часть**

• В качестве элементов контейнеров в задании 1 использовать пользовательскую структуру данных из лабы 3 (задание Г). Хотя бы для одного типа контейнера.

• Провести инициализацию контейнеров минимум четырьмя различными способами.

• При реализации типовых операций обязательно использовать итераторы нескольких типов.

• Провести тестирование скорости выполнения этих операций для всех контейнеров для большого количества элементов. Сопоставить результаты с теоретически ожидаемыми.

• При решении задания 1 обязательно использовать функции.

• Выполнить все задачи в заданиях 2 и 3

• Решить задачи из МООК

o https://stepik.org/lesson/53376/step/9

o https://stepik.org/lesson/53379/step/9?unit=31454

o

1. Создать контейнеры для хранения данных: array, vector, list, set, map, multimap, .
2. Проинициализировать контейнеры разными способами.
3. Для каждого типа контейнера выполнить следующие типовые операции (или аналогичные): вставка, поиск, удаление, изменение, копирование в другой контейнер (хотя бы один).
4. Используя итераторы
   1. Вывести результаты всех действий на экран.
   2. Сохранить и загрузить данные с использованием файлов.

Индивидуальная часть

Задание 2 – 10 Вариант.

Заполнить массив данных (vector) случайными числами в диапазоне m1-m2. Выполнить набор действий в соответствии с вариантом при помощи стандартных алгоритмов, объектов-функций и предикатов. После выполнения каждого действия выводить на экран результат.

* m1=-5, m2=11
* вычислить произведение всех элементов
* подсчитать количество чётных элементов
* удалить все нечётные элементы

**Задание 3 – 10 Вариант.**

Создать ассоциативные контейнеры в соответствии с вариантом, используя собственные способы расширения и пользовательскую структуру в качестве ключа. Заполнить контейнер данными. После выполнения каждого действия выводить на экран результат.

* контейнер map
* способы расширения: компаратор лямбда функция
* структура-ключ: дата (год, месяц, день); данные – фамилия знаменитости, родившейся в эту дату

## Краткое словесное описание алгоритмов

2-я задача:

1. Создать контейнер vector
2. Инициализировать контейнер vector элементами от -5 до 11
3. Вывести этот массив в консоль(-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 9 10 11)
4. Выполнить произведение всех элементов с помощью **accumulate(m.begin(), m.end(), 1, multiplies<int>)**
5. Вывести результат в консоль (0, т.к. умножение на 0)
6. Подсчитывание всех четных элементов с помощью **count\_if**
7. Вывод результата (8)
8. Удаление всех нечетных элементов с помощью **erase(remove\_if())**
9. Вывод результата (-4 -2 0 2 4 6 8 10)

3-я задача:

1. Создать структуру Date
2. Создать лямбда функцию компаратор, которая будет сортировать элементы контейнера map при добавлении.
3. Создать контейнер map с decltype(dateComaparator).
4. Инициализируем элементы контейнера
5. Выводим элементы контейнера в консоль.

## **Таблица спецификаций функций**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Имя функции** | **Назначение** | **Тип результата** | **Параметры** |
| **1** | dateComaparator | Лямбда функция для сравнения дат, их сортировки | auto | Data &a, Data &b |
| **2** | output\_of\_elements | Вывод элементов контейнера | - | Контейнер(vector<int>) |
| **3** | initialCont | Заполнение контейнера элементами | - | Контейнер(vector<int>) |
| **4** | speedTest | Выводит имя исследуемого контейнера в файл  Заполняет контейнеры в зависимости от их типа  Ищет минимальное число в контейнере  Добавляет число  Время всех операций записывается в файл | - | Контейнер(Type m)  ofstream &out – ссылка на файл  int\* a – массив элементов, которые используются для проверки скорости. |

## **Результаты проверки скорости инициализации элементов в разных контейнерах**

## 

Они совпадают с теоретически ожидаемыми

## **Программа на языке С++**

**1-я задача:**

<https://replit.com/@GieorghiiRasput/sixlab#main.cpp>

**2-я задача:**

<https://replit.com/@GieorghiiRasput/sixlabsecondtasksmol>

**3-я задача:**

<https://replit.com/@GieorghiiRasput/six-labthirdtasksmol>