

## Пояснительная записка

### Описание полученного задания:

Задача номер 6 (пассажирский транспорт), функция номер 11 (сортировка выбором по убыванию).

- ✓ Обобщенный артефакт, используемый в задании – Пассажирский транспорт;
- ✓ Базовые альтернативы и их уникальные параметры:
  1. *Самолет*: дальность полета – целое, грузоподъемность – целое;
  2. *Поезд*: количество вагонов – целое;
  3. *Корабль*: водоизмещение – целое, вид судна – перечислимый тип (лайнер, буксир, танкер).
- ✓ Общие для всех альтернатив переменные: скорость – целое, расстояние между пунктами отправления и назначения – действительное.
- ✓ Общая для всех альтернатив функция: идеальное время прохождения пути (действительное).
- ✓ Функция: упорядочить элементы контейнера по убыванию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

### Структура проекта:

**project/**

**tests/** - входные и выходные файлы для тестирования программы;

**main.cpp** – точка входа.

**/.../** - заголовочные файлы и файлы реализации проекта.

Заголовочные файлы проекта: random.h, airplane.h, train.h, ship.h, transport.h, container.h

Файлы реализации проекта: main.cpp, airplane.cpp, train.cpp, ship.cpp, transport.cpp, container.cpp

### Основные характеристики программы:

- Количество заголовочных файлов: 6
- Количество файлов реализации: 6
- Размер исходных файлов:  $\approx 19.5$  Кб
- Размер исполняемого файла: 102 Кб
- Время выполнения программы на различных тестовых файлах (взято усредненное значение по результатам нескольких запусков):

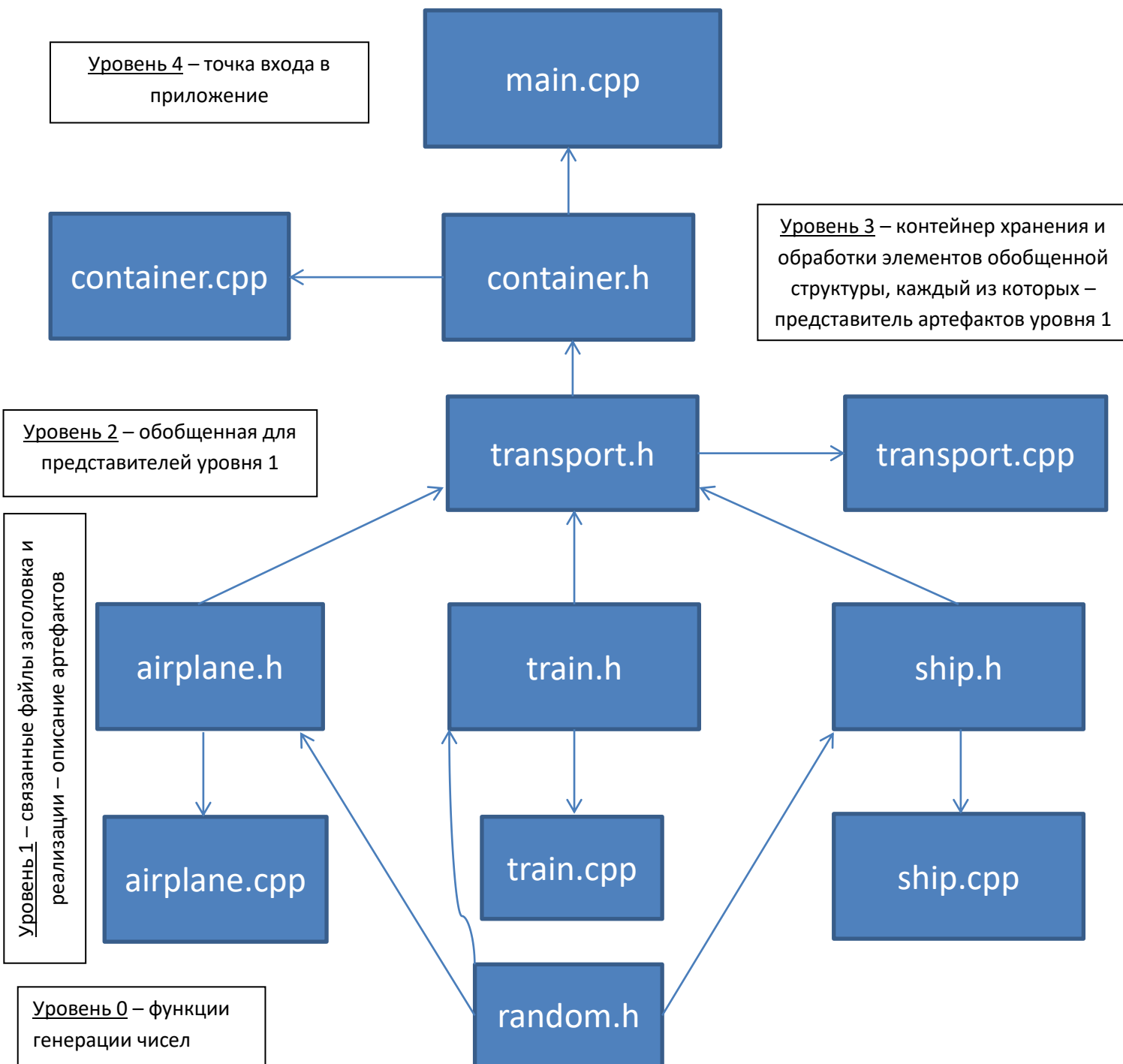
| <i>Количество элементов</i> | <i>Тип входных данных</i> | <i>Время работы (мс)</i> |
|-----------------------------|---------------------------|--------------------------|
| 0                           | Ввод с файла              | 0.0003                   |
| 2                           | Ввод с файла              | 0.00037                  |
| 4                           | Ввод с файла              | 0.000417                 |
| 8                           | Ввод с файла              | 0.000597                 |
| 10                          | Ввод с файла              | 0.000658                 |
| 10                          | Генерация данных          | 0.000638                 |
| 100                         | Генерация данных          | 0.001567                 |
| 1000                        | Генерация данных          | 0.010604                 |
| 5000                        | Генерация данных          | 0.193349                 |
| 10000                       | Генерация данных          | 0.784332                 |

Примечание к тестовым файлам: для каждого типа входных данных отведено по 5 тестов, идущих по возрастанию количества обрабатываемых элементов. В выходном файле присутствуют и несортированный, и отсортированный варианты. Пара входной-выходной файл имеет одинаковый индекс (1-5 для первого типа входных данных, 5-6 для второго типа входных данных).

### Архитектура приложения:

В разработанном программном продукте использован многоуровневый шаблон «ведущий-ведомый», при котором разные компоненты взаимодействуют друг с другом по принципу иерархии, вычисляя результаты функций на основе данных от «подчиненных» модулей.

Схема, применимая к разработанной программе:



### Инструментальные средства:

- Виртуальная машина Oracle VM VirtualBox (Linux, Ubuntu);
- Языки программирования: C/C++;
- IDE: Clion (совместно с CMake);
- Библиотеки: `stdlib.h`, `stdio.h`, `time.h`, `math.h`