**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

Архитектура вычислительных систем.

Практическое задание 2. Вариант 8.

**Исполнитель**

Студентка группы БПИ205

Баранюк И.С.

Описание полученного задания

Найти частное от деления года создания на количество символов в названии языка программирования. Присутствует 3 категории языков: процедурные, объектно-ориентированные и функциональные. Дополнительно упорядочить элементы контейнера по убыванию используя сортировку с помощью прямого обмена или пузырька (Bubble Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

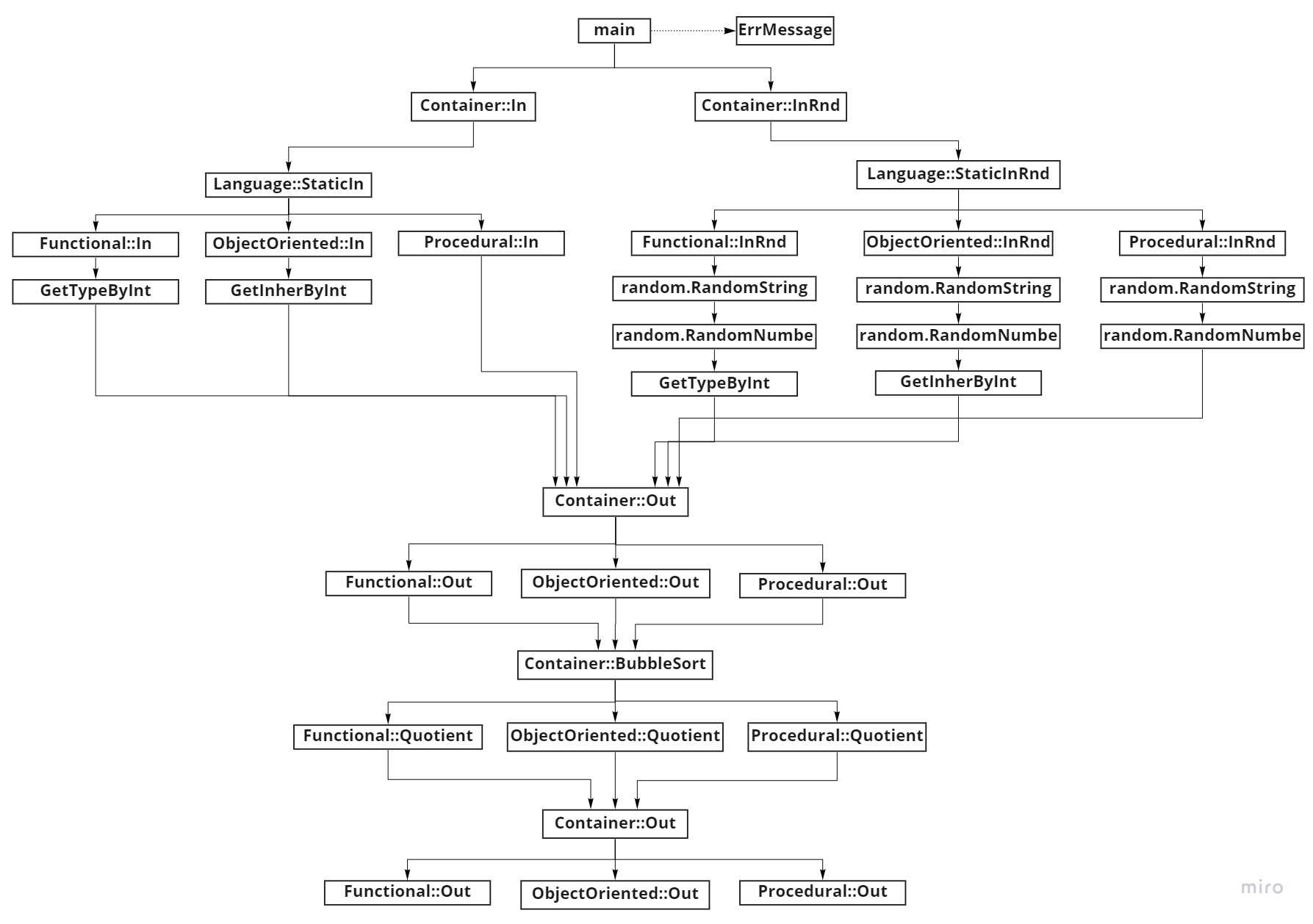
Структурная схема изучаемой архитектуры ВС

Таблица типов

|  |  |
| --- | --- |
| bool | 1 |
| string | 32 |
| int | 4 |
| float | 4 |
| enum Typing | 4[0] |
| enum Inheritance | 4[0] |
| class Functional  name: string  creation\_year: int  popularity: float  typing: Typing  lazy\_evaluation\_support: bool | 45  32[0]  4[32]  4[36]  4[40]  1[44] |
| class ObjectOriented  name: string  creation\_year: int  popularity: float  Inheritance: inheritance | 44  32[0]  4[32]  4[36]  4[40] |
| class Procedural  name: string  creation\_year: int  popularity: float  has\_abstract\_data\_types: bool | 41  32[0]  4[32]  4[36]  1[40] |
| class Language | 0 |
| class container  len: int  cont: Language\*\*  size: int | 16  4[0]  8[4]  4[12] |
| class Random  letters: char[26] | 26  26[0] |

Память программы

|  |  |
| --- | --- |
| main()  argc: int  argv: char\*\*  container: Сontainer\*  random: bool  output\_file\_name: string  size: int  input\_file: ifstream  output\_file: ofstream | 1089  4[0]  8[4]  8[12]  1[20]  32[21]  4[53]  520[57]  512[577] |
| Functional::In  input\_file: &ifstream  type: int | 8  4[0]  4[4] |
| Functional::InRnd  random: Random  type: int | 30  26[0]  4[26] |
| Random:: RandomString  new\_string: string  length: int | 36  32[0]  4[32] |
| Language::StaticInRnd  language: Language\*  random: Random  key: int | 38  8[0]  26[8]  4[34] |

Stack

Основные характеристики программы

Число интерфейсных модулей: 6

Число модулей реализации: 6

Общий размер исходных текстов: 470 строка

Время выполнения программы для различных тестовых наборов данных

**Ввод из файла:**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

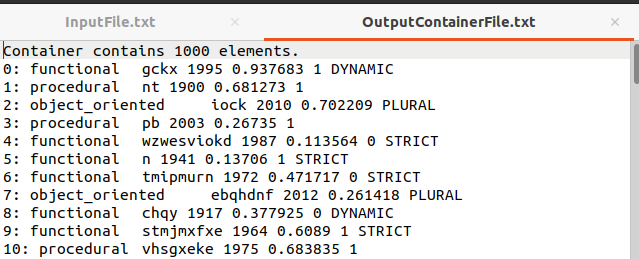
**Изображение выглядит как текст

Автоматически созданное описание**

**Случайная генерация данных:**

Изображение выглядит как текст

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Сравнительный анализ**

Разница в размере кода с предыдущим заданием небольшая.

Разница во времени исполнения показана в таблице:

|  |  |  |
| --- | --- | --- |
|  | **Статическая типизация процедурный подход** | **Статическая типизация объектно-ориентированный подход** |
| **Из файла 4 элемента** | **0,014 s** | **0,006 s** |
| **Случайная генерация 1000 элементов** | **0,023 s** | **0,038 s** |
| **Случайная генерация 9000 элементов** | **2,003 s** | **2,271 s** |

При чтении из файла программа выполняется быстрее при второй реализации, но это не очень корректно сравнивать, потому что при выполнении этого задания я использовала fstream, а в предыдущем stdio.h. Но при случайной генерации программа выполняется дольше, чем в первой реализации, даже при условии, что вывод должен происходить быстрее (исходя из сравнения времени выполнения при чтении из файла) В итоге, судя по всему, статическая типизация с объектно-ориентированным подходом работает дольше, чем с процедурным подходом.