

# Архитектура вычислительных систем

## Домашнее задание 2

### Пояснительная записка

> Вариант №288. Языки программирования и удаление элементов, для которых значение, полученное с использованием функции, общей для всех альтернатив, меньше, чем среднее арифметическое для всех элементов контейнера.

Описание файлов, содержащихся в домашней работе:

Файл запуска программы, обработки ошибок.

- main.cpp

Файлы, содержащие определение класса контейнер, реализацию его методов и функции удаления нужных элементов.

- container.h
- container.cpp

Файлы, содержащие определение абстрактного класса "Язык программирования", реализацию ее методов и функции частного (общей для всех альтернатив).

- programming\_language.h
- programming\_language.cpp

Файлы, содержащие определение и реализацию наследников класса "Язык программирования"

- procedural.cpp
- procedural.h – процедурный язык программирования (уникальное свойство – наличие/отсутствие абстрактных типов данных)
- objectoriented.cpp

- `objectoriented.h` - объектно-ориентированный язык (уникальное свойство – наследование)
- `functional.cpp`
- `functional.h` -функциональный язык (уникальные свойства – схема типизации, поддержка “ленивых” вычислений)

Файл, содержащий класс `FileReader` и реализацию его методов для считывания информации из файла

- `filereader.cpp`
- `filereader.h`

Файл, содержащий класс `StringRandomizer` и реализацию его методов для создания случайной строки (названия языка программирования)

- `StringRandomizer.cpp`
- `StringRandomizer.h`

Итого: 7 интерфейсных модулей и 8 модулей реализации.

Время выполнения программы:

Test 1

real 0m0.021s

user 0m0.003s

sys 0m0.004s

Test 2

real 0m0.010s

user 0m0.002s

sys 0m0.004s

### Test 3

real 0m0.013s  
user 0m0.002s  
sys 0m0.004s

### Test 4

real 0m0.016s  
user 0m0.003s  
sys 0m0.006s

### Test 5

real 0m0.012s  
user 0m0.003s  
sys 0m0.004s

### Random Test 1(5000)

real 0m0.060s  
user 0m0.018s  
sys 0m0.026s

### Random Test 2(333)

real 0m0.017s  
user 0m0.003s  
sys 0m0.006s

Сравнительный анализ процедурного и объектно-ориентированного подхода:

1. Благодаря объектно-ориентированному подходу кода было написано меньше, чем при процедурном.
2. Как по мне, код при объектно-ориентированном подходе стал более читаемым.
3. Но, к большому удивлению, программа стала работать медленнее, чем при процедурном подходе.

## Схема наследования в программе

