Архитектура вычислительных систем.

Вариант 293 (задание 13, функция 21)

Доржиев Донир Саянович БПИ208

1. **Описание задания**

### **Общие для всех альтернатив переменные:**

* Название – строка символов

### **Общие для всех альтернатив функции:**

Частное от деления числа гласных букв в названии на общую длину названия (действительное число)

### **Обобщенный артефакт** - растение

### **Базовые альтернативы**

1. Деревья (возраст – длинное целое)

2. Кустарники (месяц цветения – перечислимый тип)

3. Цветы (домашние, садовые, дикие… – перечислимый тип)

1. **Описание структуры ВС**

|  |  |
| --- | --- |
| **Типы данных** | |
| **Тип** | **Размер** |
| Int | 4 |
| Double | 8 |
| Char | 1 |
| FILE | 4 |

**Базовые Классы**

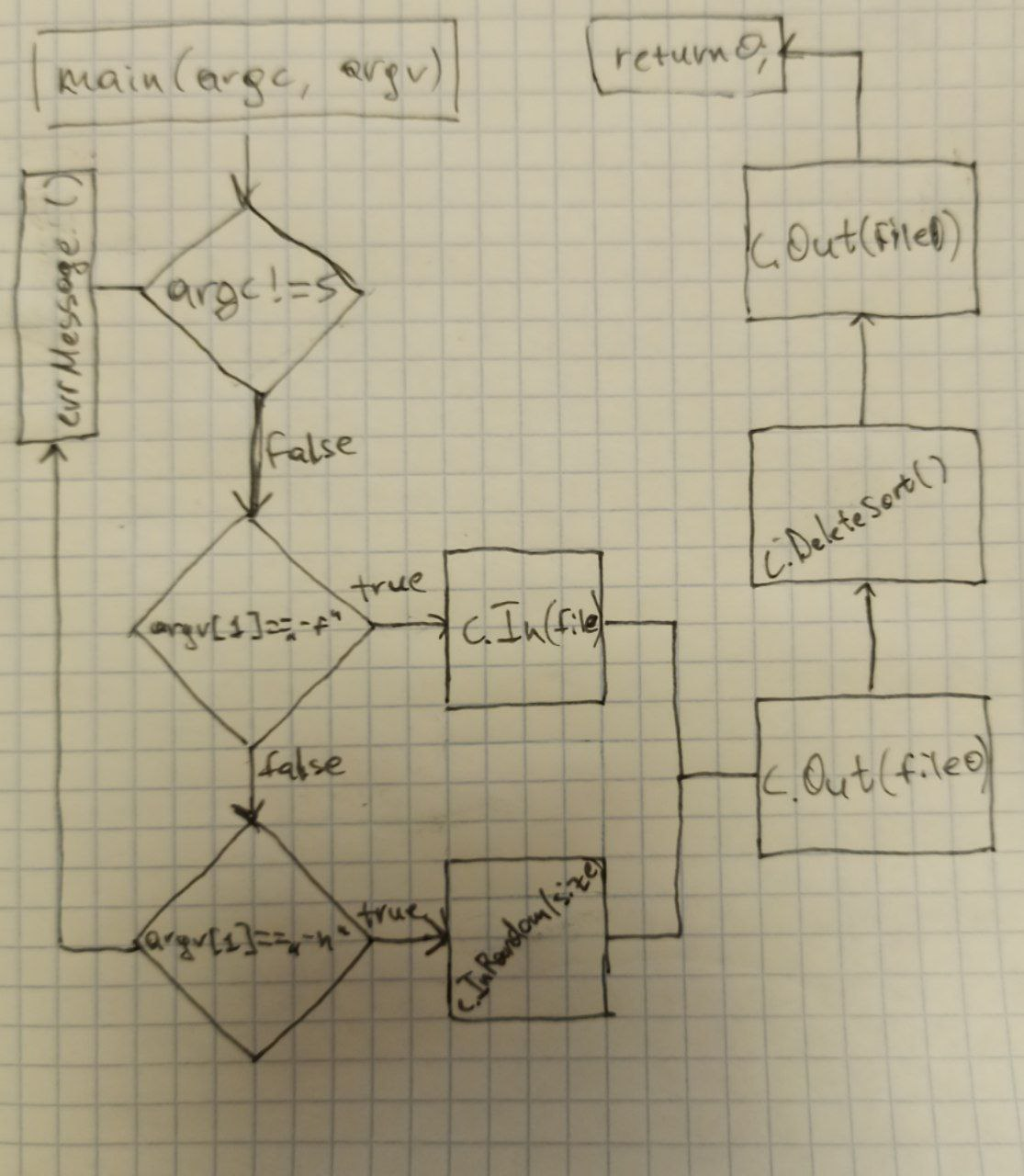
|  |  |
| --- | --- |
| **Базовые Классы** | Container [80004 байт] |
| Поля | Static: Отсутствует  Local: int length [4 байта]  plant \*cont[80000] байт |
| Методы | Static: Отсутствует  Local:  void Init();  void Clear();  void In(FILE \*file);  void InRandom(int size);  void Out(FILE \*file);  void DeleteSort() ; |
| Enum | **-** |
| **Базовые Классы** | plant [0 байт] |
| Поля | Static: Отсутствует; Local: Отсутствует |
| Методы | Static: Plant \*StaticIn(FILE \*file, int key);  Plant \*StaticInRandom();  Local: ~Plant (); |
| Enum | **-** |

**Производные классы**

|  |  |
| --- | --- |
| **Производный класс** | tree: plant[16 байт] |
| Поля | Static: -  Local: const char \*name [8 байт];  int length [4 байт];  int age [4 байт]; |
| Методы | Static: -  Local: ~tree();  void In(FILE \*fileIn);  void InRandom();  void Out(FILE \*fileOut);  double Fraction(); |
| Enum | - |
| **Производный класс** | bush: plant[20 байт] |
| Поля | Static: -  Local: const char \*name [8 байт];  Int length [4 байт];  month month\_ [4 байт]; |
| Методы | Static: -  Local: ~ bush();  void In(FILE \*file);  void InRandom();  void Out(FILE \*file);  double Fraction(); |
| Enum | month [4 байт] |
| **Производный класс** | flower: plant[20 байт] |
| Поля | Static: -  Local: const char \*name [8 байт];  Int length [4 байт];  type type\_ [4 байт]; |
| Методы | Static: -  Local: ~ flower();  void In(FILE \*file);  void InRandom();  void Out(FILE \*file);  double Fraction(); |
| Enum | type [4 байт] |

|  |  |
| --- | --- |
| Main (80016 байт) | |
| Container container | 80004 |
| Int size | 4[80004] |
| FILE input | 4[80008] |
| FILE first\_output | 4[80012] |

**Схемы:**



**Рис.1**

1. **Основные характеристики программы**

1) число интерфейсных модулей (заголовочных файлов) – 6

**plant.h, tree.h, bush.h, flower.h, container.h, random.h**

2) число модулей реализации (фалов с определением программных объектов) – 6

**plant.cpp, tree.cpp, bush.cpp, flower.cpp, container.cpp, main.cpp**

3) общий размер исходных текстов – 1+2+1+1+1+1+4+3+3+3+2+2 = 24 КБ

4) полученный размер исполняемого кода – 53 КБ

5) время выполнения программы для различных тестовых наборов данных (время в миллисекундах - ms):

1 тест:

elapsed time: 6 ms.

2 тест:

elapsed time: 8 ms.

3 тест:

elapsed time: 9 ms.

4 тест:

elapsed time: 4 ms.

5 тест:

elapsed time: 13 ms.

1. **Расположение данных**

Входные данные лежат в:

… \CSA\_HW2\cmake-build-debug\tests

Выходные лежат в:

…\CSA\_HW2\cmake-build-debug\outTests

1. **Сравнение с процедурным подходом**

* При выполнении ООП программы необходимо зачастую придумывать схему, отражающую композицию программы (из каких сущностей она состоит, есть ли между ними какие-то отношения). При процедурном подходе всего этого не требуется, что является несомненно достоинством данного подхода.
* При процедурном подходе программа выполняется немного быстрее, чем ООП программа (результаты в таблице ниже).
* Процедурный подход используется больше в небольших программах, где нет сложной и массивной архитектуры, так как он быстрее по времени. Если же требуется написать большую программу, то это лучше делать в стиле ООП, потому что читаемость кода в данном случае значительно повышается, что можно отнести к достоинству данного подхода.
* Также взаимодействие между объектами в процедурном подходе происходило через обобщенную структуру, что не совсем удобно, в этом плане ООП имеет преимущество.

|  |  |  |
| --- | --- | --- |
|  | **Первая работа** | **Вторая работа** |
| **Первый тест** | 0.0000007 s | 0.006 seconds |
| **Второй тест** | 0.0000005 s | 0.008 seconds |
| **Третий тест** | 0.0000004 s | 0.009 seconds |
| **Четвертый тест** | 0.0000006 s | 0.004 seconds |
| **Пятый тест** | 0.0000008 s | 0.013 seconds |

1. **Ввод данных**

**1**

**oak 3 54**

**2**

**barberry 8 0**

**3**

**rose 4 2**

Первая переменная – вид растения (0 – дерево, 1 – куст, 3 – цветок), вторая переменная – название растения, третья переменная – длина его имени, четвертая переменная – дополнительные параметры.

Параметры для дерева – возраст: int64\_t

Параметр для куста – переменная, которая отвечает за месяц цветения

0 – январь, 1 – февраль, 2 – март, 3 – апрель, 4 – май, 5 – июнь, 6 – июль, 7 – август, 8 – сентябрь, 9 – октябрь, 10 – ноябрь, 11 – декабрь.

Параметры для цветка – тип выращивания цветка:

0 - домашние, 1 - огородные, 2 - дикие