

Отчет

Пасилецкий Даниил Олегович

ФКН ПИ БПИ202. Вариант 55. Вариант задания 13. Вариант функции 4.

Программа выполнена на NASM. Методом компиляции программы с C++ по средствам gсс.

Программа содержит следующие классы:

1. tree - дерево содержащие имя и возраст.
2. bush - кустарник содержит имя и месяц цветения.
3. flowers - цветок содержит имя и вид.
4. plants - базовый класс, в котором используется ключ для определения типа растения. 1 - tree; 2 - bush; 3 - flowers
5. container - место для хранения массива растений.

Согласно варианту 13 задания. Так же реализована шейкерная сортировка (Shaker Sort).

Число интерфейсных модулей: 0

Число модулей реализации: 6

Размер файлов исходного кода : 37 КБ

Размер скомпилируемого кода: 2800 КБ

Запуск программы

Запуск программы производится через консоль с указанием специальных аргументов:

-f infile outfile01 outfile02 - ввод из файла *infile*, а вывод в *outfile01* - содержимое контейнера, а в *outfile02* - контейнер после сортировки.

-n number outfile01 outfile02 - создание растений рандомом, *number* - число созданных растений. Вывод в *outfile01* - содержимое контейнера, а в *outfile02* - контейнер после сортировки.

Ввод в программу через файл осуществляется следующим способом:

- В начале вводится код элемента 1 - tree; 2 - bush; 3 - flowers.
- Затем через пробел (перенос строки) имя.
- Также через пробел индивидуальная характеристика в следующем формате:
 - для дерева количество лет (целое число)
 - для кустарника месяц цветения (целое число от 1 до 12 включительно)
 - для цветка его вид, так же целое число где:
 - 1 - домашний
 - 2 - садовый
 - 3 - дикий

После всего пустая строка НЕ допускается

Вывод программы

Программа выводит файлы в 2 файла указанные при запуске, в первом файле находится не отсортированный массив, где указаны характеристики каждого растений. Во втором файле находится уже отсортированный массив.

С целью демонстрации двух способов вывода перечисления, в flowers оно выводится словами, а в bush номером месяца.

Вывод

Код на NASM работает значительно быстрее по сравнению с другими, я зыками программирование. Но он абсолютно не читаемый. И очень трудно написать. Так же очень трудна работа со строками (из-за этого после мучительных попыток написать создание рандомной строки, было принято решение использовать

компиляцию с с++). NASM подходит для того где важна скорость работы программы, например: драйвера.