ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К домашнему заданию №1 по АВС

*Сергеев Илья БПИ207*

Описание задания

Задание выполнено с целью изучения архитектуры статически типизированного процедурного языка программирования.

Условие задачи

-Обобщённый артефакт, используемый в задании – различные числа.

-Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив) - 1. Комплексные (действительная и мнимая части – пара действительных чисел) 2. Простые дроби (числитель, знаменатель – пара целых чисел) 3. Полярные координаты (угол [радиан] – действительное; координаты конечной точки на плоскости)

-Общие для всех альтернатив переменные – отсутствуют

-Общие для всех альтернатив функции - Приведение каждого значения к действительному числу, эквивалентному записанному. Например, для комплексного числа осуществляется по формуле: sqrt(d^2+i^2)), а для полярных координат - расстояние.

Обработка данных контейнера

Упорядочить элементы контейнера по убыванию используя сортировку с помощью разделения (Quick Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Тестирование

Программа обрабатывает лишь часть множества некорректного ввода. Завершение работы с ошибкой возможно при несоответствия локали (такой тест содержится в папке tests и описывает ситуацию, при которой для ввода нецелых чисел используется запятая заместо точки).

Поступающие на вход из файла входные данные корректны тогда и только тогда, когда:

1. Файл начинается с неотрицательного целого числа типа int, описывающего кол-во поступающих на вход артефактов. Если число больше нуля, то за ним следует пробел.
2. После кол-ва артефактов в файле идут по трое через пробел сначала название последующего артефакта (complex – комплексное число, polar – число в полярной с-ме координат, fraction – дробь), а затем два числа типа double или int (первая и вторая составляющая артефакта, например, в случае с дробью, знаменатель и числитель).

После ввода кол-ва элементов создаётся, заполняется и выводится в первый выходной файл контейнер для хранения артефактов. Затем он сортируется. Далее во второй выходной файл выводится список значения общей для всех альтернатив функции и отсортированный по нему контейнер.

Не обрабатываются следующие некорректные входные данные:

1. На вход помимо названий артефактов подаются не числа.
2. Значения, поступающие не приводимы к типам int (в случае с дробью) или к double (в случае с комплексным числом и числом в полярной с-ме координат)
3. Кол-во артефактов меньше первого числа в файле.
4. Для обозначения нецелых чисел используется несоответствующий настройкам локали знак (например, запятая)

Обрабатываются следующие исключения

1. Указано название несуществующего артефакта
2. Количество артефактов превышает значение первого числа в файле.

Метрики, определяющие характеристики программы для различных тестов

Число интерфейсных модулей:

Число заголовочных файлов – 5

Число модулей реализации:

Число программных файлов – 6

Общий размер исходных текстов

complex\_number.c – 983 байта, container.c – 3.17 кб, fraction\_number.c – 991 байт, main. C- 2.26 кб, number.c – 1.91 кб, polar\_number.c – 1.31 кб

complex\_number.h – 690 байт, container.h – 1.04 кб, fraction\_number.h – 701 байт, number.h – 914 байт, polar\_number – 629 байт

исполняемый файл (.exe) – 46 кб.

Время работы программы на различных тестовых данных

test\_1.txt – 1 мс, test\_2.txt (666 артефактов) – 15 мс, test\_3.txt (420 артефактов) – 13 мс, test\_5.txt – 1 мс, test\_6.txt – 1 мс, тест с 9999 объектов – 182 мс.