

Практическое занятие 15

Многофайловые приложения

Пименов Евгений Сергеевич

Курс «Программирование»

Сибирский государственный университет телекоммуникаций и информатики (Новосибирск)

Осенний семестр, 2016

Определить структуру `Fraction` – простая дробь.

Поля структуры:

- `numerator`
- `denominator`

Реализовать функции:

- `fraction_sum`
- `fraction_multiplication`

Разработать тестовое приложение для демонстрации операций.

Исходный код программы должен быть организован следующим образом:

- `Fraction.h` – определение структуры, заголовки функций
- `Fraction.c` – реализация функций
- `main.c` – точка входа в программу

Реализовать набор структур и функций для вывода геометрических примитивов в формате [WKT](#): Point, LineString, Polygon.

В заголовочных файлах размещаются определения структур и прототипы функций. В файлах `.c` – реализация функций.

Все функции, относящиеся к определенному типу, в качестве первого входного параметра принимают переменную соответствующего типа.

Функции для работы с точкой – точку, для работы с линией – линию, и т.д.

Файлы `Point.h`, `Point.c` :

- Структура `Point` : композиция двух целочисленных координат
- Функция `point_print_coordinates` : выводит координаты, `x y`
- Функция `point_print` : выводит `"POINT (x y)"`

Файлы `LineString.h`, `LineString.c` :

- Структура `LineString`: массив точек, количество точек в ломаной
- Функция `line_string_print` : `LINESTRING (x1 y1, x2 y2, x3 y3)`
- Функция `line_string_print_coordinates` : выводит координаты ломаной, `x1 y1, x2 y2, ...`
- Функция `line_string_add_point` : добавляет точку к ломаной

Файлы `Polygon.h`, `Polygon.c` :

- Структура `Polygon`: массив ломаных
- Функция `polygon_print`: `POLYGON ((...), (...), ...)`
- Функция `polygon_add_line_string`

Разработать тестовое приложение, в котором выполняется:

- Создание полигона
- Вывод полигона в формате WKT
- Вывод ломаных, составляющих полигон
- Вывод точек, составляющих полигон

Добавить функции сериализации геометрических примитивов:

```
<object>_to_string
```

Аргументы:

- [in] Объект
- [out] Строка для сохранения результата

Вывод примитивов реализовать через функции сериализации

```
man sprintf
```