Мальцева Ангелина БПИ206

Номер варианта: 214
Номер задания: 4, Номер функции: 16

ОТЧЕТНОСТЬ К ТРЕТЬЕМУ ЗАДАНИЮ

ЗАДАНИЕ

Реализовать программу, где:

1) Обобщенный класс: объемная (трехмерная) геометрическая фигура.
2) Базовые альтернативы: шар (доп. поле: целочисленный радиус), параллелепипед (доп. поле: три целочисленных ребра), правильный тетраэдр (доп. поле: длина ребра – целое)
3) Общие для всех альтернатив переменные: плотность материала фигуры (действительное число)
4) Общие для всех альтернатив функции: вычисление площади поверхности (действительное число)

Обработка всех данных происходит за счет сортировки деление пополам.

СТРУКТУРНАЯ СХЕМА ПРОГРАММЫ

Таблица типов

int	4
double	8
char	1
list	Зависит от данных
class Sphere: 1. __radius__ : Int	**4** 4[0]
class Parallelepiped: 1. __x__ : int 2. __y__ : int 3. __z__ : int	**12** 4[0] 4[4] 4[8]
class Tetrahedron: 1. __x__: int	**4** 4[0]
class Shape: 1. __density__ : double	**8** 8[0]
class Container: 1. __shapes__: list 2. size_ : int	**-** Зависит от количества фигур 4[?]

Память программы

main(): 1. start_time: time 2. container: Container 3. outputData: ofstream 4. sortedData: ofstream	**-** Зависит от количества фигур в контейнере
def readData(self, input_stream): 1. firstParam: int 2. secParam: int	**-**

3. thirdParam: int 4. density: double 5. type_of_shape: int 6. my_shape_params: list	Зависит от количества фигур в контейнере
def rndData(self, input_stream): 1. firstParam: int (не всегда) 2. secParam: int (не всегда) 3. thirdParam: int (не всегда) 4. density: double (не всегда) 5. type_of_shape: int	- Зависит от количества фигур в контейнере и их типа
def set_size(self, size): 1. size: int	**4** 4[0]

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1. Число заголовочных файлов: 0
2. Число исходных файлов: 7
3. Общий размер исходных текстов: 9.572 КБ
4. Размер исполняемого файла: 3.17 КБ
5. Время работы на тестах:
   test1: 0.003c
   test2: 0.005c
   test3: 0.015c
   test4: 0.429c
   test5: 22.882c

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА

Это третья работа по курсу «Архитектура вычислительный систем». Суть задания (в сравнении с другими) не изменилась, однако изменилась реализация.

Теперь код написан на Python с использованием динамической типизации. Python это интерпретируемый язык, так что не обязательно объявлять типа переменных и сами переменные, если мы их только собираемся использовать (можно объявлять непосредственно перед использованием). Это сильно затрудняет положение при подсчете занимаемой памяти. Создан базовый класс Shape, у которого есть три наследника Sphere, Parallelepiped и Tetrahedron.

Общий размер исходных файлов сильно меньше, чем в двух предыдущий реализациях программы, так как кода стало меньше, потому что была использована динамическая типизация и пропала нужда в реализации заголовочных файлов.

В Python отсутствует файл .exe, вместо исполняемого файла можно использовать любой файл .py (в данном случае main.py). Следовательно размер исполняемого файла уменьшился до размера обычного файла.

Время работы на первых четырех тестов у данной реализации меньше, но на пятом тесте (при генерации рандомных объектов в количестве 10000 штук) во времени она сильно проигрывает.

Можно сделать вывод, что реализация программы на Python хорошо работает с маленькими данными. При обработке больший данных лучше использовать одну из предыдущих реализаций.