ОПИСАНИЕ ЗАДАНИЯ

Разработать программный продукт с использованием динамической проверки типов во время выполнения программы. Программа должна содержать следующие структуры и функции:

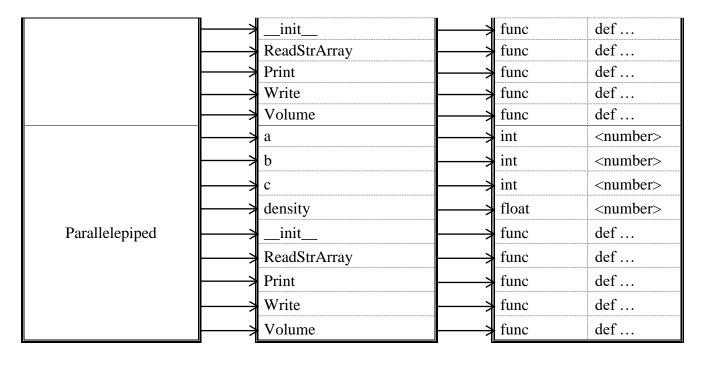
структуры и футкци	1	T	1
Обобщенный артефакт,	Базовые	Общие для всех	Общие для всех
используемый	альтернативы	альтернатив	альтернатив функции
в задании	(уникальные	переменные	
	параметры,		
	задающие		
	отличительные		
	признаки		
	альтернатив)		
Объемная (трехмерная)	1. Шар	Плотность материала	Вычисление объема
геометрическая фигура.	(целочисленный	фигуры	(действительное
	радиус)	(действительное	число)
	2. Параллелепипед	число)	
	(три целочисленных		
	ребра)		
	3. Правильный		
	тетраэдр (длина		
	ребра – целое)		

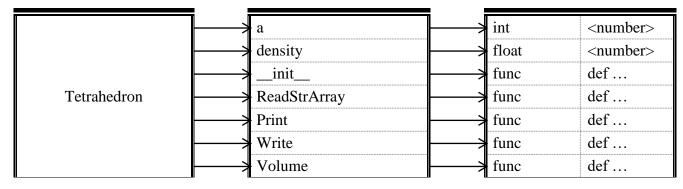
Дополнительная функция:

Упорядочить элементы контейнера по убыванию используя сортировку с помощью прямого слияния (Straight Merge). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив (объём).

СТРУКТУРНАЯ СХЕМА АРХИТЕКТУРЫ ВС С ПРОГРАММОЙ:

Таблица классов	Таблица имен	Описание
Container	storeinit Write Print InRnd Volume merge_list merge_sort	list
Shape	init ReadStrArray Print Write Volume radius	func def func def func def func def func def func def
Ball	density	int <number> float <number></number></number>





ОТРАЖЕНИЕ НА ПАМЯТЬ МЕТОДОВ КЛАССОВ:

Память программы	
main.py	
main.py(StartBaseLogic)	
ReadStrArray	

inputFileName	
outputFileName	
ifile	
str	
strArray	
count	
container.py	
container	
ofile	
container	
strArray	
i	
key	
density	

Память данных	
list	[]
list	[]
file	fileName
list	[]
list	[]
Int	<number></number>
module	extender.py
Container	container.py
file	fileName
Container	container.py
list	[]
Int	<number></number>
Int	<number></number>
Int	<number></number>
Float	<number></number>

Container. Merge_list Merge_sort	

shape
arrayLen
figNum
str
self
mid
left
right
i
j
k

Shape	shape.py
Ball	ball.py
Parallelepiped	parallelepiped.py
Tetrahedron	tetrahedron.py
int	<number></number>
int	<number></number>
string	<string></string>
Container	container.py
int	<number></number>
Shape	shape.py
Shape	shape.py
int	<number></number>
int	<number></number>
Int	<number></number>

ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

- Число заголовочных файлов 1 (extender.py)
- Число модулей реализации 7
- Общий размер исходных текстов 326 строк кода вместе с комментариями
- Размер исполняемого файла -0 (файл не генерируется, так как питон скриптовый язык).
- Время выполнения программы для различных тестовых прогонов:

Номер теста	Время выполнения в секундах
Тест1	0.000998
Тест2	0.0079464
Тест3	0.012875
Тест4	0.008526
Тест5	0.020942

СРАВНЕНИЕ С ПРЕДЫДУЩИМИ ВЕРСИЯМИ ПРОГРАММЫ:

Программа работает быстрее, чем версия, написанная на C и примерно с то же скоростью, что и программа, написанная на C++:

Таблица тестов программы на С:

Номер теста	Время выполнения, сек
1	0.011
2	0.016
3	0.045
4	0.047
5	0.1

Таблица тех же тестов программы на С++:

Номер теста	Время выполнения, сек
1	0.006
2	0.008
3	0.013
4	0.017
5	0.029

- В программе появились новая опции –отпала необходимость явно задавать выходной файл (если выходной файл не задан, то программа сгенерирует его сама);
- Вывод в файл теперь осуществляется в один файл, а не в 2, как это было раньше. Это помогает немного экономить время на очистки потока и запросе файла у системы.
- Пропала возможность измерять размер типов. В Python нет возможности получить размер типа (только объекта какого-то типа), поэтому провести разумный анализ затрачиваемой памяти не представляется возможным.
- Количество строк кода уменьшилось в 2 раза, что связано с облегченным синтаксисом питона.