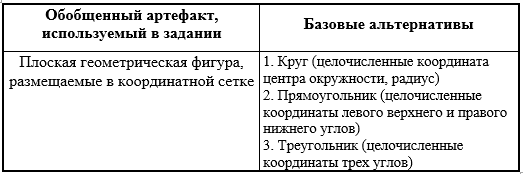
**ОПИСАНИЕ ЗАДАНИЯ (задание 1, дополнительная функция 4):**

Разработать программный продукт с использованием *динамической проверки типов во время выполнения (стиль написания – произвольный)*. Программа должна содержать следующие структуры:



Для всех альтернатив общей переменной является **цвет** (перечислимый тип). Он может принимать значения: красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый. Общей функция всех альтернатив выступает вычисление периметра фигуры (действительное число). В качестве дополнительной функции контейнера необходимо реализовать Shaker Sort.

Также нужно: разработать тестовые входные данные и провести тестирование и отладку программы на этих данных (при необходимости, программа должна правильно обрабатывать переполнение по данным); описать структуру используемой ВС с наложением на нее обобщенной схемы разработанной программы; зафиксировать количество заголовочных файлов, программных файлов, общий размер исходных текстов, полученный размер исполняемого кода (если он формируется), время выполнения программы для различных тестовых наборов данных.

**СТРУКТУРНАЯ СХЕМА АРХИТЕКТУРЫ ВС С ПРОГРАММОЙ:**

Программа разработана в 64 битной системе (дистрибутив Ubuntu на ярде Linux). Разработка велась на языке Python версии 3.9 с использованием объектно-ориентированной парадигмы программирования (с исключением для main.py). Рассмотрим все классы разработанной программы и опишем 5 функций, на примере которых отобразим графически архитектуру виртуальной Python машины с наложением на нее обобщенной схемы разработанной программы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица классов |  | Таблица имен |  | Описание | |
| Container |  | store |  | list | […] |
|  | \_\_init\_\_ |  | func | def … |
|  | Write |  | func | def … |
|  | Square |  | func | def … |
|  | Sort |  | func | def … |
|  | RemoveAlternatives |  | func | def … |
| Shape |  | color |  | Enum | class … |
|  | \_\_init\_\_ |  | func | def … |
|  | SetColor |  | func | def … |
|  | GetColorString |  | func | def … |
|  | ReadStrArray |  | func | def … |
|  | Print |  | func | def … |
|  | Write |  | func | def … |
|  | Square |  | func | def … |
| Circle |  | r |  | int | <number> |
|  | center |  | Point | class … |
|  | color |  | Enum | class … |
|  | \_\_init\_\_ |  | func | def … |
|  | SetColor |  | func | def … |
|  | GetColorString |  | func | def … |
|  | ReadStrArray |  | func | def … |
|  | Print |  | func | def … |
|  | Write |  | func | def … |
|  | Square |  | func | def … |
| Rectangle |  | point1 |  | Point | class … |
|  | point2 |  | Point | class … |
|  | color |  | Enum | class … |
|  | \_\_init\_\_ |  | func | def … |
|  | SetColor |  | func | def … |
|  | GetColorString |  | func | def … |
|  | ReadStrArray |  | func | def … |
|  | Print |  | func | def … |
|  | Write |  | func | def … |
|  | Square |  | func | def … |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица классов |  | Таблица имен |  | Описание | |
| Triangle |  | a |  | Point | class … |
|  | b |  | Point | class … |
|  | c |  | Point | class … |
|  | color |  | Enum | class … |
|  | \_\_init\_\_ |  | func | def … |
|  | SetColor |  | func | def … |
|  | GetColorString |  | func | def … |
|  | ReadStrArray |  | func | def … |
|  | Print |  | func | def … |
|  | Write |  | func | def … |
|  | Square |  | func | def … |
| Point |  | x |  | Int | <number> |
|  | y |  | Int | <number> |
| Exceptions |  | IncorrectArguments |  | func | def … |
|  | IncorrectCount |  | func | def … |
|  | IncorrectFileInput |  | func | def … |
| GeneratorAndWriter |  | RndGenerateArr |  | func | def … |
|  | GetCordsForTriangl |  | func | def … |
|  | GetCordsForRectangle |  | func | def … |
|  | GetCordsForCircle |  | func | def … |
|  | ConinueOrExit |  | func | def … |
|  | ReadStrArray |  | func | def … |
|  | WriteTest |  | func | def … |

**ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:**

* Число заголовочных файлов – 1 (extender.py)
* Число модулей реализации – 10
* Общий размер исходных текстов – 21 315 байт
* 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 | 0.000999 |
| Тест2 | 0.001000 |
| Тест3 | 0.009978 |
| Тест4 | 0.001000 |
| Тест5 | 0.009897 |
|  |  |