Пояснительная записка

**1. Выполнил:**

Тарковский Роман Гасанович, БПИ202

**2. Описание полученного задания:**

2.1 Вариант задания: 65

2.2 Номер задачи: 9

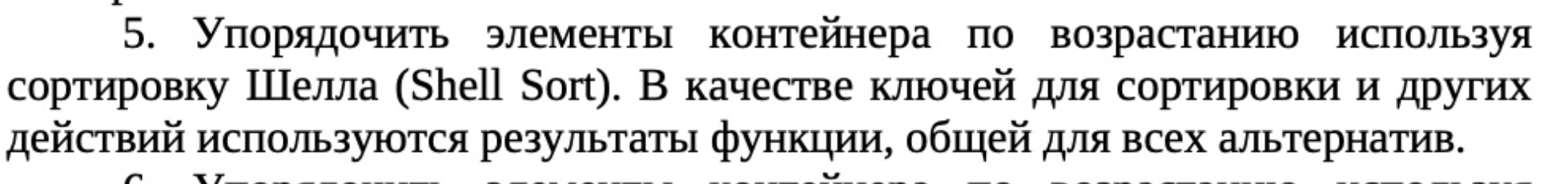
Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

2.3 Номер функции: 5



**3. Структурная схема изучаемой архитектуры ВС с размещенной на ней разработанной программой:**

1. Отображение на память содержимого модуля main

**Память программы**

**Память данных**

**Таблица имён**

figNum

outputFileName2

main.py

int

string: “…”

str

file: fileName

ifile

string: “…”

inputFileName

string: “…”

string: “…”

outputFileName

float

module: container.py

container

Start\_time

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| container.py |  |  |  |
| \_\_init\_\_ |  | func | def… |
|  |  |  |  |
| def write() |  | func | def… |
|  | shape | module | Shape.py |
| def rnd\_cont() |  | func | def… |
| def shell() |  | func | def… |
|  | store | [] | <shape> |
| Def print() |  | func | def |
| shape.py |  |  |  |
| def read\_str\_array() |  | func | def… |
|  |  |  |  |
| def write() |  | func | def… |
|  |  |  |  |
| def print() |  | func | def… |
|  |  |  |  |
| def quotient() |  | func | def… |
|  |  |  |  |
| readStrArray.py |  |  |  |
| read\_str\_array() |  | func | def… |
|  | array\_len | int | <number> |
|  | i | int | <number> |
|  | key | int | <number> |
|  | name | string | “…” |
|  | shape | module | shape.py |
| symbols\_encryption.py |  |  |  |
| \_\_init\_\_ |  | func | def… |
|  | symbols | list | [] |
| read\_str\_array() |  | func | def… |
|  | text | list | [] |
| write() |  | func | def… |
|  |  |  |  |
| quotient() |  | func | def… |
|  | En\_text | list | [] |
| print () |  | func | def… |
|  |  |  |  |
| number\_encryption.py |  |  |  |
| \_\_init\_\_ |  | func | def… |
|  | numbers | list | [] |
| read\_str\_array() |  | func | def… |
|  | text | list | [] |
| write() |  | func | def… |
|  | en\_text | list | [] |
| print() |  | func | def… |
|  |  |  |  |
| quotient() |  | func | def… |
| cyclic\_encryption.py |  |  |  |
| \_\_init\_\_ |  | func | def… |
|  | n | int | <number> |
| read\_str\_array() |  | func | def… |
|  | text | list | [] |
| write() |  | func | def… |
|  | en\_text | list | [] |
| print() |  | func | def… |
|  |  |  |  |
| quotient() |  | func | def… |

**4. Требуемые метрики, определяющие характеристики программы, для различных тестовых прогонов:**

|  |  |
| --- | --- |
| **Количество параметров, N** | **Время, S** |
| 100 | 0.48 |
| 1000 | 462 |
| 10000 | бесконечно |

**5. Данные, демонстрирующие сравнения с характеристиками ранее разработанных программ:**

5.1

5.1.1 В сравнении по времени работы со старой программой –

Очень большая разница, хоть и сложность сортировки не изменилась, сама программа работает долго при больших данных.

5.1.2 В сравнении со старой программой, где была статическая типизация, здесь же типизация динамическая, это очень удобно, если программа генерирует входные параметры сама, так как в таком случае невозможно получить TypeError, иначе нужно все время проверять входные данные, так как получить exception очень легко.

5.2 Характеристики первой разработанной программы

5.2.1 Общий размер проекта: 1,2 Мб

5.2.2 Число программных файлов проекта: 9

5.2.3 # Sources (6):

shape.cpp

cyclic\_encryption.py

symb\_encryption.py

number\_encryption.py

main.py

container.cpp