**Структура программы изучаемой ВС.**

Отображение содержимого классов Container, Text, Digits.

|  |  |  |
| --- | --- | --- |
| **Таблица классов** | **Таблица имён** | **Описание** |
| Container | \_\_init\_\_  write  quotient  bubble\_sort | func def …  func def …  func def …  func def … |
| Text | read\_array  write  encrypt  quotient  \_\_gt\_\_ | func def …  func def …  func def …  func def …  func def … |
| Digits | \_\_init\_\_  read\_array  write  encrypt  quotient  \_\_gt\_\_ | func def …  func def …  func def …  func def …  func def …  func def … |

Классы Shift и Swap я не расписываю, так как они идентичны классу Digits.

Отображение на память содержимого методов классов.

|  |  |  |
| --- | --- | --- |
| **Память программы** | **Таблица имён** | **Память данных** |
| main.py | qualifier  input\_file\_name  outpur\_file\_name  outpur\_file\_name\_sorted  in\_file  strArray  container  figNum  out\_file  out\_file\_sorted | string “…”  string “…”  string “…”  string “…”  file fileName  list […]  module container.py  int <number>  file fileName  file filename |
| Container.\_\_init\_\_ | store | list […] |
| Container.write | out\_stream  store  text | file fileName  list […]  Text object |
| Container.quotient | quotient  store  text | float <number>  list […]  Text object |
| Container.bubble\_sort | n  i  store | int <number>  int <number>  list […] |
| Digits.\_\_init\_\_ | open\_text  encrypted\_text  length | string “…”  string “…”  int <number> |
| Digits.read\_array | i  str\_array  open\_text | int <number>  list […]  string “…” |
| Digits.write | out\_stream  open\_text  encrypted\_text  quotient() | file fileName  int <number>  string “…”  func def … |
| Digits.encrypt | length  open\_text  i  isdigit()  encrypted\_text  \_\_str\_\_() | int <number>  string “…”  int <number>  func def …  string “…”  func def … |
| Digits.quotient | code\_sum  i  open\_text  len()  ord() | int <number>  int <number>  string “…”  func def …  func def … |
| Digits.\_\_gt\_\_ | quotient() | func def … |

Использование памяти показано на примере классов Container и Digits и модуля main, рассмотрение других классов не имеет смысла, так как они либо делают почти то же, что и класс Digits, либо являются условными для поддержания иерархии понятий.

Стек вызовов (сверху вниз).

main

randomize (optional)

randomize done

read\_array

read\_array done

write

write done

bubble\_sort

bubble\_sort done

write

write done

main done