|  |  |
| --- | --- |
| Типы данных | |
| class shape  color : colors | 4  4[0] |
| class rectangle : shape  x\_upper\_left: int  y\_upper\_left: int  x\_lower\_right: int  y\_lower\_right: int | 16  4[0]  4[4]  4[8]  4[12] |
| class triangle : shape  x1: int  x2: int  x3: int  y1: int  y2: int  y3: int | 24  4[0]  4[4]  4[8]  4[12]  4[16]  4[20] |
| class circle : shape  x\_center: int  y\_center: int  radius: int | 12  4[0]  4[4]  4[8] |
| class container  length: int  cont: shape\*\* | 80004  4[0]  8 \* 10000 = 80000[4]  (однако это только память, затрачиваемая на указатели на объекты классов геометрических форм, помимо них также будет выделена память на куче под каждый из инициализированных объектов) |
| Enum colors {  red, orange,  yellow, green,  blue, | 4  4[0] |
| int | 4[0] |
| double | 8[0] |

|  |  |
| --- | --- |
| Глобальная память | |
| container:: MAX\_LENGTH: int | 4[0] |

|  |
| --- |
| Таблица виртуальных функций |
| shape::in(FILE \*fptr) -> void |
| shape::in\_rnd()-> void |
| shape::output(FILE \*fptr) -> void |
| shape::perimeter() -> double |
| rectangle::in(FILE \*fptr) -> void |
| rectangle::in\_rnd()-> void |
| rectangle::output(FILE \*fptr) -> void |
| rectangle::perimeter() -> double |
| triangle::in(FILE \*fptr) -> void |
| triangle::in\_rnd()-> void |
| triangle::output(FILE \*fptr) -> void |
| triangle::perimeter() -> double |
| circle::in(FILE \*fptr) -> void |
| circle::in\_rnd()-> void |
| circle::output(FILE \*fptr) -> void |
| circle::perimeter() -> double |

|  |
| --- |
| Глобальная память функций |
| shape::generic\_init\_input(FILE \*fptr) -> shape\* |
| shape::generic\_init\_rand() -> shape\* |
| shape::colors\_to\_string(int index) -> char\* |

|  |  |
| --- | --- |
| Память программы | |
| container::shaker\_sort(container &c) ->void  c: container  arr: shape\*  control: int  left: int  right: int  i: int | 8[0]  8[8]  4[16]  4[20]  4[24]  4[28] |
| container::perimeter\_sum(container &c) -> double  c: &container  sum: double  i: int | 8[0]  8[8]  4[12] |
| shape \*in\_rnd()  sp: shape\*  k: int | 8[0]  4[8] |
| static shape::generic\_init\_input(FILE \*fptr) -> shape\*  fptr: FILE\*  type: int  color: int  object: shape\* | 8[0]  4[8]  4[12]  8[20] |