Типы данных

char	1	
int	4	
double	8	
class Rectangle		
x_l_t	int	4 [0]
y_l_t	int	4 [4]
x_r_b	int	4 [8]
y_r_b	int	4 [12]
		16
class Triangle		
x1	int	4 [0]
x2	int	4 [4]
y1	int	4 [8]
y2	int	4 [12]
z1	int	4 [16]
z2	int	4 [20]
		24
class Circle		
x_c	int	4 [0]
y_c	int	4 [4]
r	int	4 [8]
		12
class Random		
first	int	4 [0]
last	int	4 [4]
		8
class Shape		
rnd20	Random	8 [0]
rnd3	Random	8 [8]
rnd7	Random	8 [16]
color	int	4 [24]
		28
class Container		
len	int	4 [0]
storage [10000]	Shape*	80000 [4]
		80004

Память в программе

void Container::In(FILE *file)		
file	FILE*	8 [0]
k	int	
storage	Shape*	
len	int	
		8
void Container::InRnd(int size)		
size	int	4 [0]
len	int	
storage	Shape*	

		4
void Container::ColorPrint(int n, FILE *file)		
n	int	4 [0]
file	*FILE	8 [4]
		12
Shape *Shape::StaticIn(FILE *file, int k)		
file	FILE*	8 [0]
k	int	4 [8]
sp	Shape*	8 [12]
color	int	
		20
Shape *Shape::StaticInRnd()		
k	int	4 [0]
color	int	4 [4]
sp	Shape*	8 [8]
		16

Схема работы

```
int main(int argc, char* argv[])
```

```
argc != 5
```

```
printf("Start")
```

```
void errMessage1()
```

```
!strcmp(argv[1], "-f")
```

```
file = fopen(argv[2], "r")
void Container::In(FILE *file)
```

```
!strcmp(argv[1], "-n")
```

```
Shape *Shape::StaticIn(FILE *file, int k)
```

```
void errMessage2()
```

```
auto size = atoi(argv[2])
```

```
size < 1 || size > 10000
```

```
c.InRnd(size)
```

```
errMessage3
```

```
storage[len] = Shape::StaticInRnd()
    Shape *Shape::StaticInRnd()
```

```
   auto k = Shape::rnd3.Get()
auto color = Shape::rnd7.Get()
              int Get()
```

```
file1 = fopen(argv[3], "w")
fprintf(file1, "Filled container:\n")
c.Out(file1)
```

```
void Container::Out(FILE* file)
```

```
        file2 = fopen(argv[4], "w")
          c.ProcessingVar23(file2)
void Container::ProcessingVar23(FILE* file)
```

```
printf("\nFinished")
```