

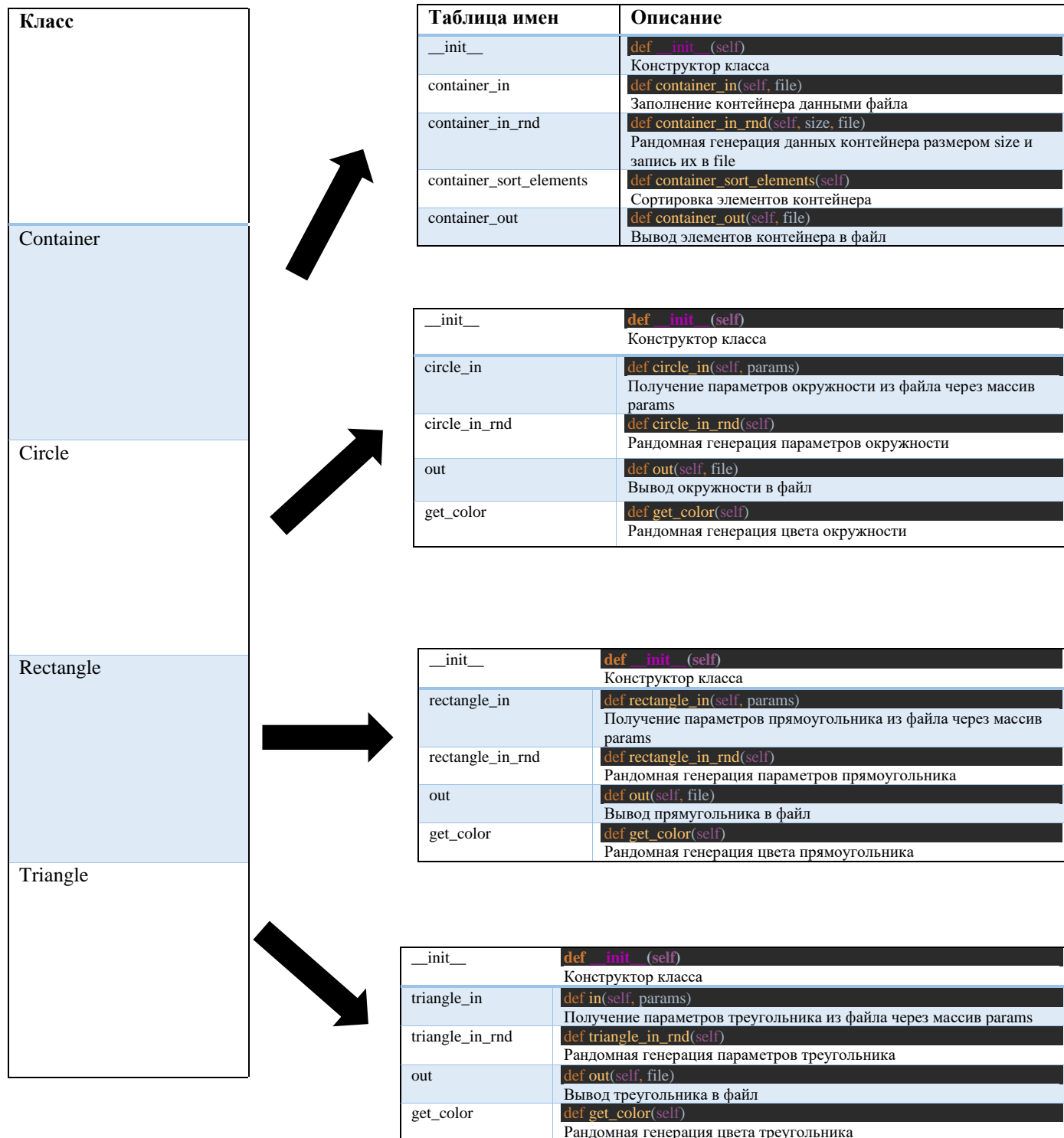
Отчет о программе, разработанной на Python

Вариант 2. Функция 10

Демиденко Ксения БПИ208

Структура программы

Содержимое классов



Память методов модулей

Память программы	Таблица имен	Память данных	
main.py	input_file	file	fileName
	output_file	file	fileName
	container	module	container.py
	size	int	<number>
container.py			
def __init__	size	int	<number>
	content	list	[...]
def container_in	params	list	[...]
	str_type	string	"..."
	c	module	circle.py
	r	module	rectangle.py
def container_in_rnd	t	module	triangle.py
	shape_type	int	<number>
	c	module	circle.py
	r	module	triangle.py
def sort_elements	t	module	rectangle.py
def container_out			
circle.py			
def __init__	self.x	int	<number>
	self.y	int	<number>
	self.rad	int	<number>
	self.perimeter	double	<double>
	self.color	string	"..."
def circle_in			
def circle_in_rnd			
def out	output	string	"..."
def get_color	color_int	int	<number>
rectangle.py			
def __init__	self.x_up_left	int	<number>
	self.y_up_left	int	<number>
	self.x_down_right	int	<number>
	self.y_down_right	int	<number>
	self.perimeter	double	<double>
	self.color	string	"..."
def rectangle_in			
def rectangle_in_rnd			
def out	output	string	"..."
def get_color	color_int	int	<number>
triangle.py			
def __init__	self.x1	int	<number>
	self.y1	int	<number>
	self.x2	int	<number>
	self.y2	int	<number>
	self.x3	int	<number>
	self.y3	int	<number>
	self.perimeter	double	<double>
def triangle_in	self.color	string	"..."
def trinagle_in_rnd			
def side			
def out	output	string	"..."
def get_color	color_int	int	<number>

Спецификация

Спецификация ВС

OS: macOS Big Sur 11.5.2

Спецификация средств разработки

IDE: PyCharm 2021.1 (Community Edition)

Характеристики проекта

Количество программных объектов (.py): 5

Размер исходных файлов

Файл	Размер (Б)	Размер на диске (КБ)
main.py	1 111	4
container.py	2 544	4
circle.py	1557	4
rectangle.py	2082	4
triangle.py	2939	4

Время выполнения программы в сравнении с программой на C++

Тест	Количество тестовых наборов данных	Время(с.)	Время(с.) работы программы на C++ на тестовых наборах с таким же количеством данных
test1.txt	3	0.00125	0.000356
test2.txt	10	0.00092	0.000401
test2.txt	1000	0.25574	0.014865
test2.txt	5000	4.988	0.207818
test2.txt	10000	18.3425	0.775573

Мы видим, что программа, написанная на языке Python, работает в разы медленнее (особенно это заметно на больших наборах тестовых данных). Поэтому программа на C++ имеет огромное преимущество по времени работы. Плюсом Python же является его простота, а также динамическая типизация, которая позволяет избежать множества ошибок.

Ввод данных

Для работы с данными из существующего файла

`python main.py <имя_входного_файла> <имя_выходного_файла>`

Для случайной генерации данных

`python main.py <имя_файла_куда_запишутся_данные> <имя_выходного_файла> <количество_объектов>`

Формат данных во входном файле

`<n- количество объектов>`

На следующих *n* строках записаны объекты в виде:

-для круга:

`<CIRCLE> <x центра> <y центра> <r радиус> <COLOR (RED, ORANGE, etc)>`

-для прямоугольника:

`<RECTANGLE> <x левой верхней> <y левой верхней> <x правой нижней> <y правой нижней> <COLOR (RED, ORANGE, etc)>`

-для треугольника:

`<TRIANGLE> <x 1-ой> <y 1-ой> <x 2-ой> <y 2-ой> <x 3-ей> <y 3-ей> <COLOR (RED, ORANGE, etc)>`