Отчет о программе, разработанной на Python Вариант 2. Функция 10 Демиденко Ксения БПИ208

Структура программы Содержимое классов

Класс

Circle

Triangle

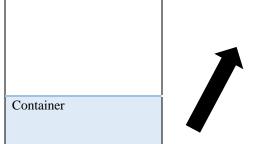


Таблица имен	Описание
init	definit(self) Конструктор класса
container_in	def container_in(self, file) Заполнение контейнера данными файла
container_in_rnd	def container_in_rnd(self, size, file) Рандомная генерация данных контейнера размером size и запись их в file
container_sort_elements	def container_sort_elements(self) Сортировка элементов контейнера
container_out	def container_out(self, file) Вывод элементов контейнера в файл



init	def init (self) Конструктор класса
circle_in	def circle_in(self, params) Получение параметров окружности из файла через массив params
circle_in_rnd	def circle_in_rnd(self) Рандомная генерация параметров окружности
out	def out(self, file) Вывод окружности в файл
get_color	def get_color(self) Рандомная генерация цвета окружности



init	definit(self) Конструктор класса
rectangle_in	def rectangle_in(self, params) Получение параметров прямоугольника из файла через массив params
rectangle_in_rnd	def rectangle_in_rnd(self) Рандомная генерация параметров прямоугольника
out	def out(self, file) Вывод прямугольника в файл
get_color	def get_color(self) Рандомная генерация цвета прямоугольника



init	definit(self)
	Конструктор класса
triangle_in	def in(self, params)
	Получение параметров треугольника из файла через массив params
triangle_in_rnd	<pre>def triangle_in_rnd(self)</pre>
	Рандомная генерация параметров треугольника
out	def out(self, file)
	Вывод треугольника в файл
get_color	def get_color(self)
	Рандомная генерация цвета треугольника

Память методов модулей

Память мегодов модулеи Память программы	Таблица имен	Память д	іянных
main.py	input_file	file	fileName
mum.py	output_file	file	fileName
	container	module	container.py
	size	int	<number></number>
container.py	5322		
definit	size	int	<number></number>
	content	list	[]
def container_in	params	list	
_	str_type	string	[] ""
	c	module	circle.py
	r	module	rectangle.py
	t	module	triangle.py
def container_in_rnd	shape_type	int	<number></number>
	С	module	circle.py
	r	module	triangle.py
	t	module	rectangle.py
def sort_elements			
def container_out			
circle.py			
definit	self.x	int	<number></number>
	self.y	int	<number></number>
	self.rad	int	<number></number>
	self.perimeter	double	<double></double>
	self.color	string	" "
def circle_in			
def circle_in_rnd			
def out	output	string	" "
def get_color	color_int	int	<number></number>
rectangle.py			
definit	self.x_up_left	int	<number></number>
	self.y_up_left	int	<number></number>
	self.x_down_right	int	<number></number>
	self.y_down_right	int	<number></number>
	self.perimeter	double	<double></double>
	self.color	string	دد ۲۰ ۰۰۰
def rectangle_in			
def rectangle_in_rnd			
def out	output	string	""
def get_color	color_int	int	<number></number>
triangle.py			
definit	self.x1	int	<number></number>
	self.y1	int	<number></number>
	self.x2	int	<number></number>
	self.y2	int	<number></number>
	self.x3	int	<number></number>
	self.y3	int	<number></number>
	self.perimeter	double	<double></double>
	self.color	string	""
def triangle_in			
def trinagle_in_rnd			
def side			
def out	output	string	"…"
def get_color	color_int	int	<number></number>

Спецификация

Спецификация ВС

OS: masOS Big Sur 11.5.2

Спецификация средств разработки IDE: PyCharm 2021.1 (Community Edition)

Характеристики проекта

Количество программных объектов (.ру): 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 С++

Тест	Количество тестовых наборов данных	Время(с.)	Время(с.) работы программы на C++ на
	наооров данных		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 <uмя_входного_файла> <uмя_выходного_файла>

Для рандомной генерации данных

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

Формат данных во входном файле

<п- количество объектов>

На следующих п строках записаны объекты в виде:

-для круга:

<CIRCLE> <x центра> <y центра> <r paduyc> <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-o $\check{u}>$ <y 1-o $\check{u}>$ <x 2-o $\check{u}>$ <y 2-o $\check{u}>$ <x 3-e $\check{u}>$ <y 3-e $\check{u}>$ <COLOR (RED, ORANGE, etc)>