Белова Наталья Александровна
БПИ-204
Вариант: 10
Функция: 9

СТРУКТУРА ПРОГРАММЫ

Содержимое классов

Таблица классов
Container
Aphorism
Proverb
Puzzle

Таблица имен	Описание
__init__	`def __init__(self)` Конструктор класса
fill_randomly	`fill_randomly(self, count)` Рандомное заполнение контейера.
get_random_string	`def get_random_string(self, is_content=True)` Генерация строки
write	`def write(self, output_file)` Ввод в файл.
read	`def read(self, input_file)` Считывание данных из фала.

Таблица имен	Описание
__init__	`def __init__(self, author, content)` Конструктор класса
to_string	`def to_string(self)` Представление объекта в виде строки.

Таблица имен	Описание
__init__	`def __init__(self, country, content)` Конструктор класса
to_string	`def to_string(self)` Представление объекта в виде строки.

Таблица имен	Описание
__init__	`def __init__(self, answer, content)` Конструктор класса
to_string	`def to_string(self)` Представление объекта в виде строки.

Таблица классов
WellOfWisdom
Sorting

Таблица имен	Описание
__init__	`def __init__(self, content)` Конструктор класса
get_quotient	`def get_quotient(self)` Функция для получения частного от деления кол-ва знаков препинания на длину строки.

Таблица имен	Описание
merge_sort	`def merge_sort(array)` Сортировка массива

Память методов

Память программы
main.py
Container.fill_randomly
Container.read
WellOfWisdom.get_quotient

Таблица имен	Тип переменной
count	int
c	Container
output_file	file
input_file	file

Таблица имен	Тип переменной
type_of_object	int
well_of_wisdom	WellOfWisdom

Таблица имен	Тип переменной
count	int
type_of_object	string
content	string
type_content	string
well_of_wisdom	WellOfWisdom

Таблица имен	Тип переменной
counter	int

Память программы
Sorting.merge

Таблица имен	Тип переменной
left	list
right	list
n, m, k	int
C	list

СПЕЦИФИКАЦИЯ
Спецификация ВС
Operating System: macOS Big Sur
Спецификация средств разработки
IDE: PyCharm CE (v2021.1.2)

ХАРАКТЕРИСТИКИ ПРОЕКТА

Количество заголовочных файлов: 6
Количество программных объектов: 6

Общий размер исходных файлов

Файл	**Размер (в байтах)**
Aphorism.py	443
Container.py	2048
main.py	2048
Proverb.py	450
Puzzle.py	440
Sorting.py	578
WellOfWisdom.py	403

ВРЕМЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ ДЛЯ РАЗЛИЧНЫХ ТЕСТОВ

Имя файла	**Время выполнения**
test1	0.0016 с.
test2	0.00036 с.
test3	0.00085 с.
test4	0.00121 с.
test5	0.00087 с.
test6	0.0011 с.

ВВОД ДАННЫХ

Генерация рандомных объектов
-r <количество объектов> <выходной файл>

Чтение данных из файла
<входной файл> <выходной файл>

Формат данных во входном файле
<n - количество объектов>
На следующих n строках описаны объекты в следующем формате:
<тип объекта>
<текстовое содержание объекта, общее для всех альтернатив>
<уникалное текстовое содержане объекта>
Тип объекта - число от 0 до 2, гдде 0 - афоризм, 1 - пословица, 2 - загадка.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ

Сравним результаты прохождения различных тестов программы, написанной на языке C++ и Python.

Имя файла	**Время выполнения программы на языке Python**	**Время выполнения программы на языке C++**
test1	0.0016 с.	0.000296 с.
test2	0.00036 с.	0.000437 с.
test3	0.00085 с.	0.00013 с.
test4	0.00121 с.	0.000308 с.
test5	0.00087 с.	0.000173 с.
test6	0.0011 с.	0.000127 с.

Из таблицы видно, что программа на Python работает медленнее. Плюсом является простота языка. Динамическая типизация позволяет избежать множества фатальных ошибок в работе программы.