

Задание 3: Архитектура ВС с динамической типизацией

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
10. Кладезь мудрости.	1. Афоризмы (один из авторов – строка символов) 2. Пословицы и поговорки (страна – строка символов) 3. Загадки (ответ – строка символов)	Содержание кладези мудрости – строка символов	Частное от деления количества знаков препинания в содержательной строке на длину строки (действительное число)

Упорядочить элементы контейнера по возрастанию используя сортировку с помощью прямого включения (Straight Insertion). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

- 1) Структурная схема
 Файлы программы:
 __main__.py
 Aphorism.py
 Proverb.py
 Riddle.py
 WisdomGenerator.py
 WisdomSorter.py

Память Программы	Таблица имен	Память данных	
__main__.py	args	<list>	[...]
	start	datetime.datetime	datetime.datetime(2021, 11, ..., 597993)
	finish	datetime.datetime	datetime.datetime(2021, 11, ..., 597993)
	container	<list>	[...]
	inputFile	_io.BufferedReader	fileName
	wisdomType	string	"..."
	title	string	"..."
	content	string	"..."
	output1	_io.BufferedReader	fileName
	output2	_io.BufferedReader	fileName
	n	int	<number>
	i	int	<number>
Aphorism.py			
def __init__(self, title, text)	self	Aphorism	<Aphorism>
	title	string	"..."
	text	string	"..."
	counter	float	<number>
	i	int	<number>
	characteristic	float	<number>
	typename	string	"..."

Proverb.py			
def __init__(self, title, text)	self	Proverb	<Proverb>
	title	string	"..."
	text	string	"..."
	counter	float	<number>
	i	int	<number>
	characteristic	float	<number>
	typename	string	"..."
Riddle.py			
def __init__(self, title, text)	self	Riddle	<Riddle>
	title	string	"..."
	text	string	"..."
	counter	float	<number>
	i	int	<number>
	characteristic	float	<number>
	typename	string	"..."
WisdomGenerator.py	kAlphabet	string	"..."
def generateWisdom()	title	string	"..."
	content	string	"..."
	titleLength	int	<number>
	contentLength	int	<number>
	wisdomType	int	<number>
	i	int	<number>
WisdomSorter.py			
def StraightSort(container)	container	<list>	[...]
	i	int	<number>
	index	int	<number>

2) Метрики

В качестве основной метрики, определяющей характеристики программы будем использовать среднее арифметическое время работы программы на 10 000 случайных «Мудростях» (будем выполнять 5 тестов). Также будем сравнивать суммарный размер текстов.

3) Замеры характеристик

1. Процедурный подход

Среднее арифметическое: 129 430 384.2 microseconds

Суммарный размер текстов: 12839 Б

2. Объектно-ориентированный подход

Среднее арифметическое: 25 399 469.6 microseconds

Суммарный размер текстов: 7954 Б

3. Язык с динамической типизацией

Среднее арифметическое: 592 623.25 microseconds

Суммарный размер текстов: 2320 Б

Вывод: Решение более эффективно благодаря оптимизации подсчета характеристик