

Домашнее задание №2

Колтунов Кирилл, БПИ207

1. Описание задания:

Разработать программный продукт с использованием динамической проверки типов во время выполнения. Программа должна содержать следующие структуры и функции:

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
Растения	1. Деревья (возраст – длинное целое) 2. Кустарники (месяц цветения – перечислимый тип) 3. Цветы (домашние, садовые, дикие... – перечислимый тип)	Название – строка символов.	Частное от деления числа гласных букв в названии на общую длину названия (действительное число)

Дополнительная функция:

Упорядочить элементы контейнера по убыванию используя сортировку с помощью прямого слияния (Straight Merge). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив (процент гласных букв).

2. Структурная схема архитектуры ВС с программой:

Программа разработана в 64 битной системе (дистрибутив Ubuntu на ядре Linux) на языке Python версии 3.9 с использованием образного представления понятий без их явного выделения. Рассмотрим архитектуру подробнее:

Таблица представлений	Таблица имен	Описание	
Container	Read	func	def ...
	Random	func	def ...
	Print	func	def ...
	Write	func	def ...
	PercantageOfVowelLetters	func	def ...
	MergeSort	func	def ...
Flower	flowerType	Enum	class ...
	Read	func	def ...
	RandomName	func	def ...
	Random	func	def ...
	Print	func	def ...
	Write	func	def ...
	PercantageOfVowelLetters	func	def ...
	flower	list	[...]
Shrub	monthOfFlowering	Enum	class ...
	Read	func	def ...
	RandomName	func	def ...
	Random	func	def ...
	Print	func	def ...
	Write	func	def ...
	PercantageOfVowelLetters	func	def ...
	shrub	list	[...]
Tree			
	Read	func	def ...
	RandomName	func	def ...
	Random	func	def ...
	Print	func	def ...
	Write	func	def ...
	PercantageOfVowelLetters	func	def ...
	tree	list	[...]

3. Сравнительный анализ

Процедурный подход подразумевает написание программного кода без использования объектов. Процедурное программирование заключается в написании кода **с или без подпрограмм**. В процедурной программе модули взаимодействуют посредством состояния чтения и записи, которое хранится в общих структурах данных.

В мире ООП объекты являются основным предметом интереса. Объект состоит из данных и кода, которому разрешено воздействовать на эти данные, и они очень тесно связаны. Это концепция инкапсуляции, скрытия информации. **Главным преимуществом, которое используется в моей программе – полиморфизм** (способность объекта использовать методы производного класса, который не существует на момент создания базового), к сожалению, аналог чего-то подобного попросту отсутствует при процедурном подходе. Это значительно упрощает код и его понимание становится более прозрачным.

Отличие статической типизации от динамической. **Статические языки проверяют типы в программе во время компиляции**, еще до запуска программы. Любая программа, в которой типы нарушают правила языка, считается некорректной. **Динамически типизированные языки не требуют указывать тип, но и не определяют его сами**. Типы переменных неизвестны до того момента, когда у них есть конкретные значения при запуске.

В данном задании мы работали с динамической типизацией, можно сделать следующие выводы:

1. Программа стала работать заметно медленнее, а связано это с особенностями работы виртуальной машины Python (наличие дополнительного слоя – интерпретатор), а также со спецификой языка с динамической типизацией (требуется время на связывание).
2. Благодаря удобству языка Python количество кода сократилось в 2 раза, что делает его более читабельным, простым и очевидным, а также не заставляет программиста задумываться о работе с памятью, ее очисткой, указателями и прочем.
3. Добавлен вывод в консоль для удобства пользователя.

4. Основные характеристики программы:

- Количество заголовочных файлов: 1
- Количество модулей реализации: 5
- Общий размер исходных текстов: 428 строк
- Размер исполняемого кода: 20,1КБ
- Время выполнения программы для различных тестовых прогонов:

Номер теста	Время выполнения, сек
1	0.0009
2	0.0106
3	0.4105
4	0.0015
5	0.0073