Архитектура вычислительных систем.

Вариант 293 (задание 13, функция 18)

Доржиев Донир Саянович БПИ208

1. **Описание задания**

### **Общие для всех альтернатив переменные:**

1. Название – строка символов,
2. Длина имени - целое число

### **Общие для всех альтернатив функции:**

Частное от деления числа гласных букв в названии на общую длину названия (действительное число)

### **Обобщенный класс:**

Растение

### **Базовые альтернативы:**

1. Деревья (возраст – длинное целое) 2. Кустарники (месяц цветения – перечислимый тип) 3. Цветы (домашние, садовые, дикие… – перечислимый тип)

1. **Схемы:**
   * + 1. Отображение на память содержимого модуля main:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| main.py | inputFileName | string | “…” |
| outputFileName | string | “…” |
| ifile | file | fileName |
| arrayLen | int | <number> |
| Data | string | “…” |
| strArray | list | […] |
| plantNum | int | <number> |
| ofile | file | fileName |
| container | Container | container.py |
| container.py | module | extender.py |

2. Отображение на память методов класса Container:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def \_\_init\_\_ | self | Container | container.py |
| store | list | […] |
| def Print | self | Container | container.py |
| plants | list | […] |
| plant | Plant | Plant.py |
| def Write | self | Container | container.py |
| ostream | file | fileName |
| plant | Plant | Plant.py |
| plants | list | […] |
| def DeleteSort | self | Container | container.py |
| sum\_of\_fractions | float | <value> |
| plant | Plant | Plant.py |
| new\_cont | list | […] |
| new\_plant | Plant | Plant.py |

1. Отображение на память методов класса Plant:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def ReadStrArray | self | Plant | Plant.py |
| strArray | list | […] |
| i | int | <number> |
| def RandomRead | self | Plant | Plant.py |
| def Print | self | Plant | Plant.py |
| def Write | self | Plant | Plant.py |
| out\_stream | file | fileName |
| def Fraction | self | Plant | Plant.py |

1. Отображение на память методов класса Tree:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def \_\_init\_\_ | self | Tree | Tree.py |
| string | “…” | name |
| length | Int | <number> |
| age | Int | <number> |
| def ReadStrArray | self | Tree | Tree.py |
| strarray | list | […] |
| i | int | <number> |
| name | string | “…” |
| length | int | <number> |
| age | int | <number |
| def RandomRead | self | Tree | Tree.py |
| name | string | “…” |
| length | int | <number> |
| age | int | <number> |
| def Print | self | Tree | Tree.py |
| def Write | self | Tree | Tree.py |
| def Fraction | self | Tree | Tree.py |
| vowels\_count | int | <number> |

1. Отображение на память методов класса Bush:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def \_\_init\_\_ | self | Bush | Bush.py |
| name | string | “…” |
| length | Int | <number> |
| month | Enum | <number> |
| i | int | <number> |
| def ReadStrArray | self | Bush | Bush.py |
| strarray | list | […] |
| i | int | <number> |
| name | string | “…” |
| length | int | <number> |
| month | Enum | <number> |
| def RandomRead | self | Bush | Bush.py |
| name | string | “…” |
| length | int | <number> |
| month | Enum | <number> |
| def Print | self | Bush | Bush.py |
| def Write | self | Bush | Bush.py |
| def Fraction | self | Bush | Bush.py |
| vowels\_count | int | <number> |

1. Отображение на память методов класса Flower:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def \_\_init\_\_ | self | Flower | Flower.py |
| name | string | “…” |
| length | int | <number> |
| type | Enum | <number> |
| i | int | <number> |
| def ReadStrArray | self | Flower | Flower.py |
| string\_array | list | […] |
| i | int | <number> |
| name | string | “…” |
| length | int | <number> |
| type | Enum | <number> |
| def RandomRead | self | Flower | Flower.py |
| name | string | “…” |
| length | int | <number> |
| type | Enum | <number> |
| def Print | self | Flower | Flower.py |
| def Write | self | Flower | Flower.py |
| def Fraction | self | Flower | Flower.py |
| vowels\_count | int | <number> |

1. Отображение на память методов класса Randomizer:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def generate |  |  |  |
| def namegenerate | length | int | <number> |
| i | int | <number> |
| name | string | “…” |

1. Отображение на память содержимого модуля readStringArray:

|  |  |  |  |
| --- | --- | --- | --- |
| Память программы | Таблица имён | Память данных | |
| def ReadStrArray | container | Container | container.py |
| strArray | list | […] |
| arraylen | int | <number> |
| i | int | <number> |
| plantNum | int | <number> |
| def RandomRead | container | Container | container.py |
| plantNum | int | <number> |
| arraylen | int | <number> |

1. Отображение содержимого классов:

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица классов | Таблица имён | Описание | |
| Container | \_\_init\_\_ | func | def … |
| Print | func | def … |
| Write | func | def … |
| DeleteSort | func | def … |
| Plant | \_\_init\_\_ | func | def … |
| ReadStrArray | func | def … |
| RandomRead | func | def … |
| Print | func | def … |
| Write | func | def … |
| Fraction | func | def … |
| Tree | \_\_init\_\_ | func | def … |
| ReadStrArray | func | def … |
| RandomRead | func | def … |
| Print | func | def … |
| Write | func | def … |
| Fraction | func | def … |
| Bush | \_\_init\_\_ | func | def … |
| ReadStrArray | func | def … |
| RandomRead | func | def … |
| Print | func | def … |
| Write | func | def … |
| Fraction | func | def … |
| Flower | \_\_init\_\_ | func | def … |
| ReadStrArray | func | def … |
| RandomRead | func | def … |
| Print | func | def … |
| Write | func | def … |
| Fraction | func | def … |
| Randomizer | generate | func | def … |
| namegenerate | func | def … |

1. **Основные характеристики программы**

1) Число интерфейсных модулей (заголовочных файлов) – нет из-за ненадобности

2) Число модулей реализации (фалов с определением программных объектов) – 9:  
 Bush.py  
 container.py  
 extender.py  
 Flower.py  
 main.py  
 Plant.py  
 Randomizer.py  
 readStrArray.py  
 Tree.py

3) Общий размер исходных текстов – 2+2+1+2+2+1+1+3+2= 16 Кб

4) Время выполнения программы для различных тестовых наборов данных:

1 тест: 0.00200 seconds

2 тест: 0.00123 seconds

3 тест: 0.00099 seconds

4 тест: 0.00227 seconds

5 тест: 0.00569 seconds

1. **Ввод данных**

**1**

**oak 3 54**

**2**

**barberry 8 0**

**3**

**rose 4 2**

Первая переменная – вид растения (1 – дерево, 2 – кустарник, 3 – цветок), вторая переменная – за название растения, третья переменная – длина имени, четвертая – дополнительные параметры.

Параметры для дерева – это возраст: целое число

Параметр для кустарника – месяц цветения:

0 – January

1 – February

2 – March

3 – April

4 – May

5 – June

6 – July

7 – August

8 – September

9 – October

10 – November

11 - December

Параметры для цветка – тип выращивания:

0 – home

1 – garden

2 – wild

1. **Разница между динамической типизацией и процедурной, объектно-ориентированной**
2. Динамическая типизация на Python работает медленнее, чем программы со статической типизацией.
3. Взглянув на программу, мы явно не можем сразу понять какой тип используется, потому что python динамически типизированный язык, что усложняет читаемость.
4. Тем не менее чтение данных в динамически типизированном языке легче, чем в предыдущих заданиях со статической типизацией
5. Не требуется в динамической типизации использовать интерфейсные модули (.h)

**6.Вывод:**

При динамической типизации программа работает медленнее и в этом уступает процедурной и ООП парадигмам, но в преимущество данного подхода можно отнести компактность кода. В связи с тем, что программа достаточно большая, то в качестве произвольного подхода был выбран ООП подход.