Архитектура вычислительных систем.

Вариант 250 (12,18)

Выполнила: Караваева Екатерина Aндреевна

БПИ207

1. **Описание задания**

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

1. Название – строка символов,
2. Вес в граммах (целое)

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

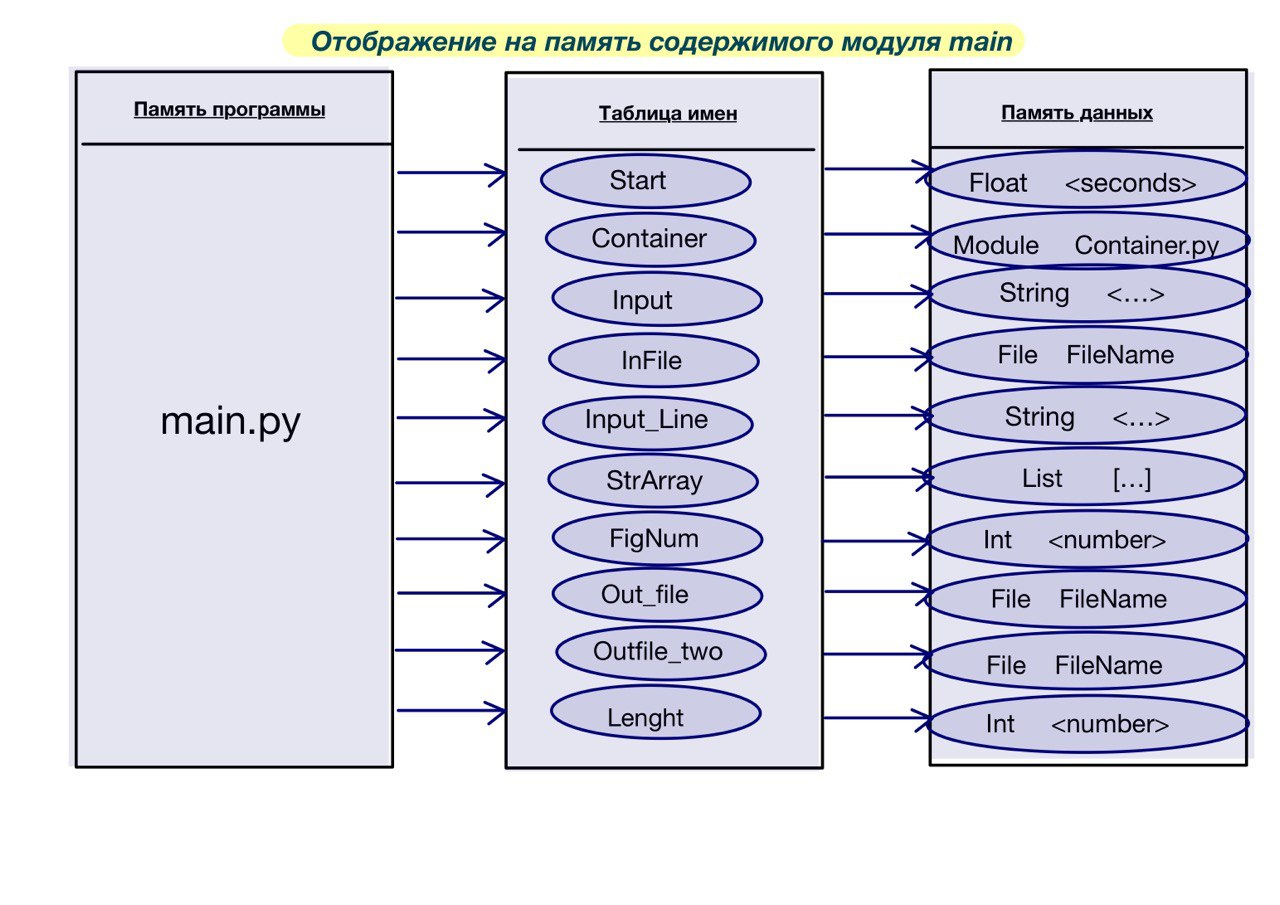
Частное от деления суммы кодов незашифрованной строки на вес (действительное число)

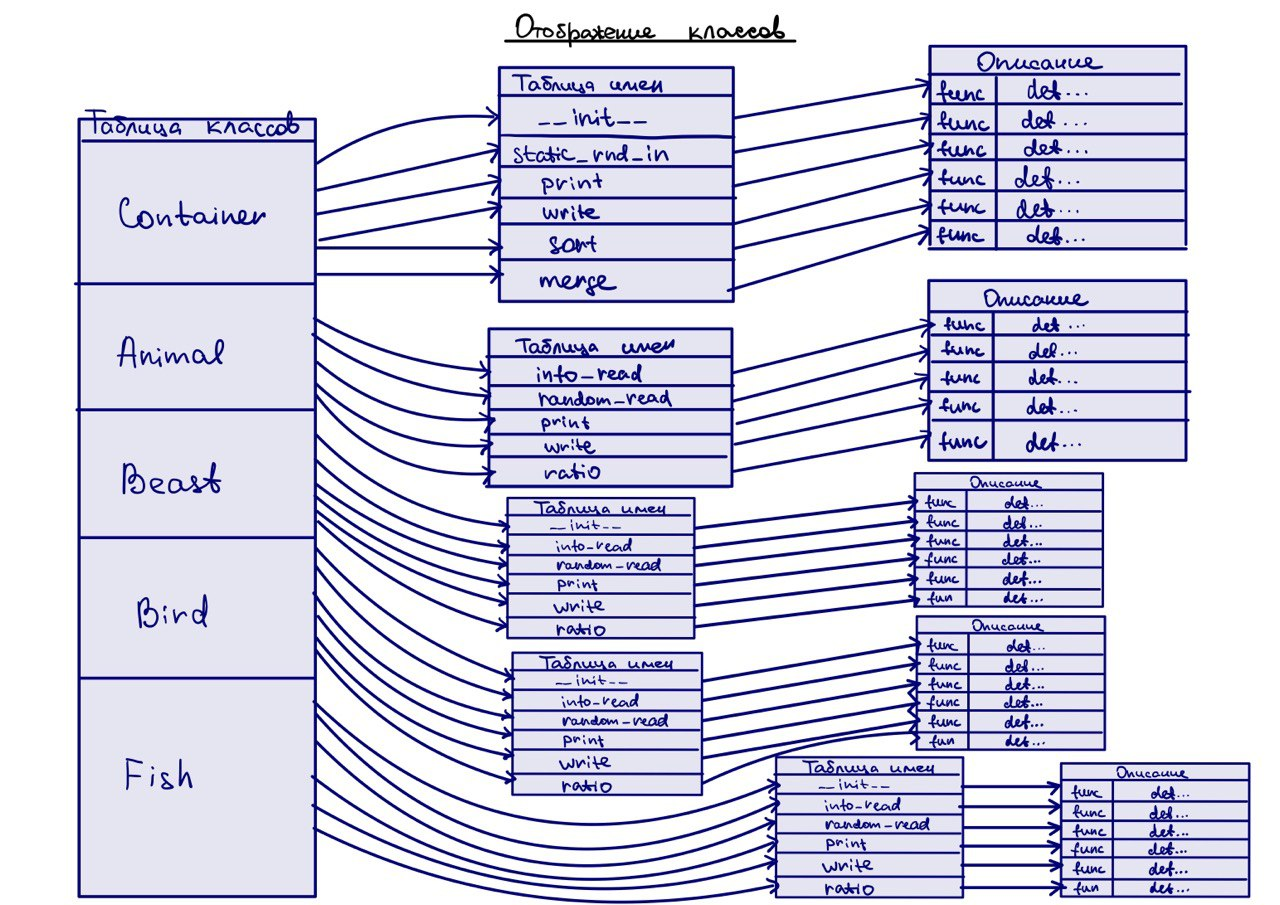
### **Обобщенный артефакт** - животные

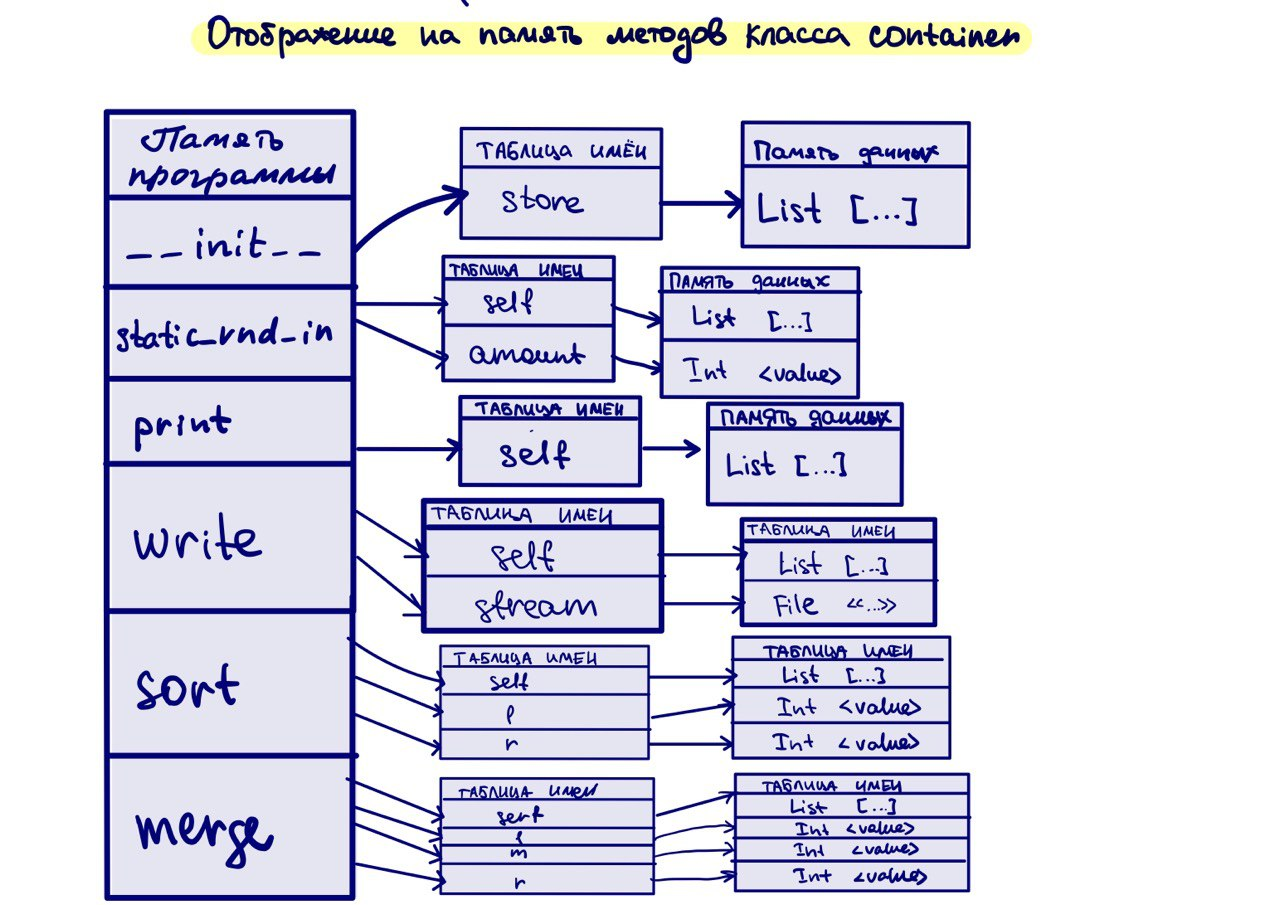
### **Базовые альтернативы**

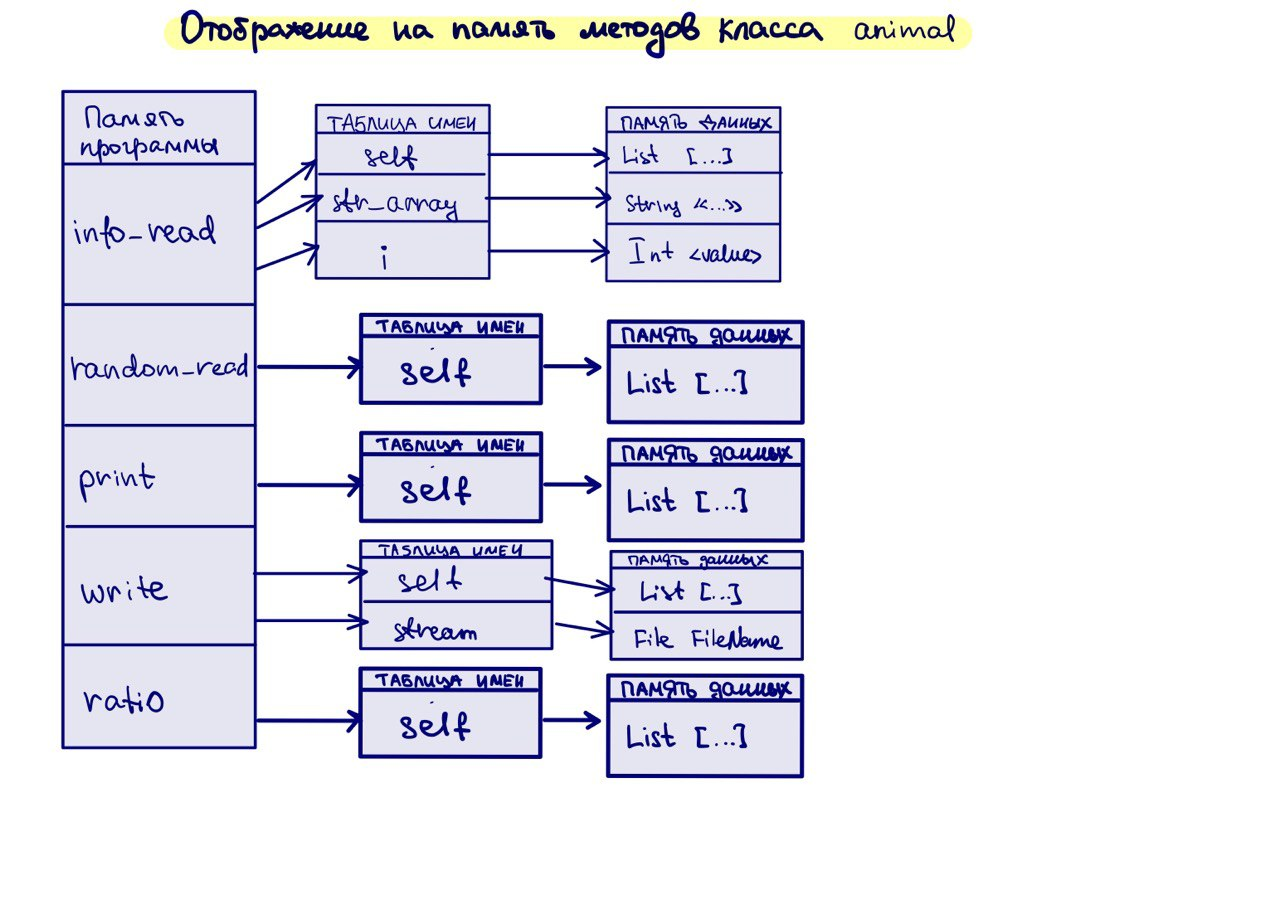
1. Рыбы (место проживания – перечислимый тип: река, море, озеро…) 2. Птицы (отношение к перелету: перелетные, остающиеся на зимовку – булевская величина) 3. Звери (хищники, травоядные, насекомоядные… – перечислимый тип)

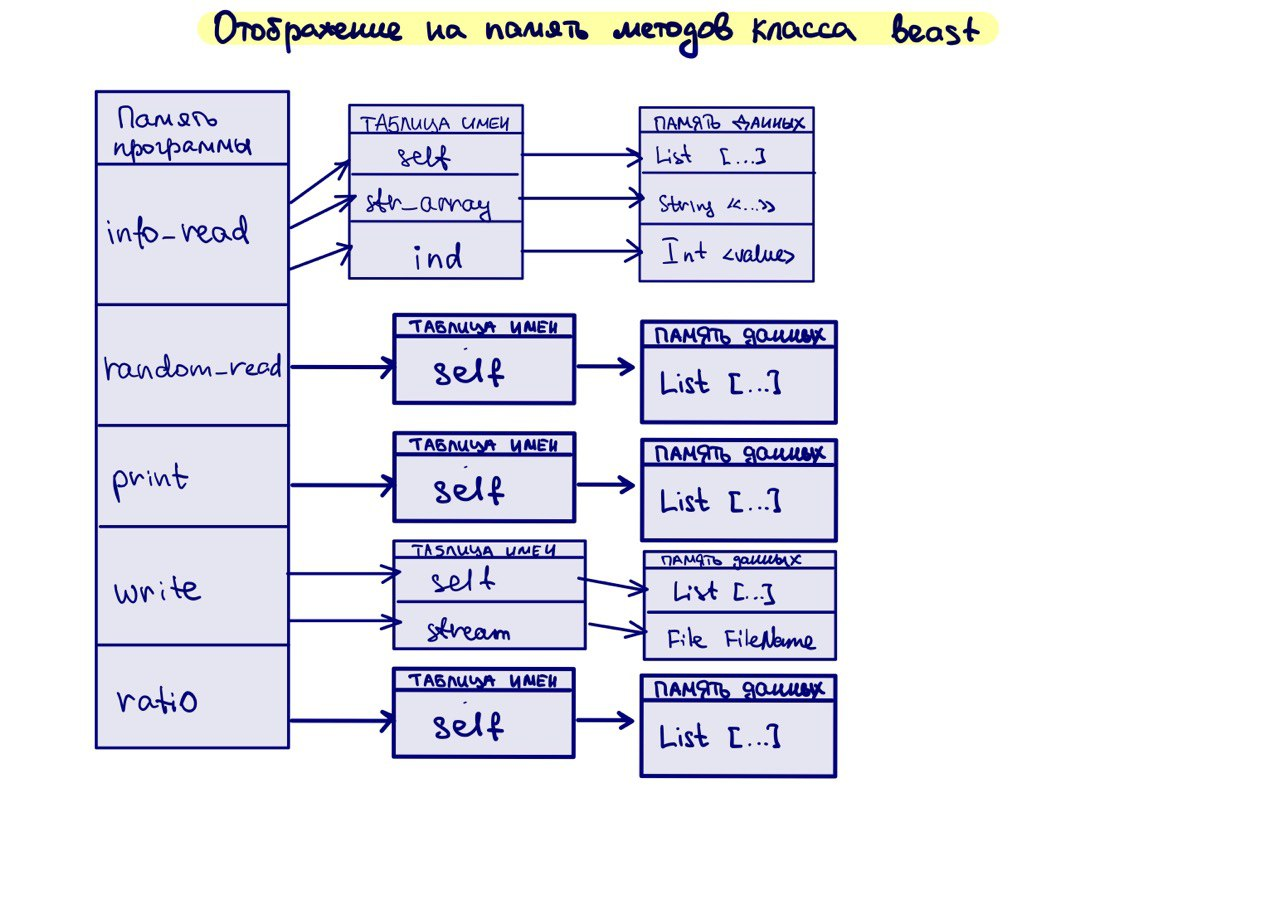
1. **Схемы:**

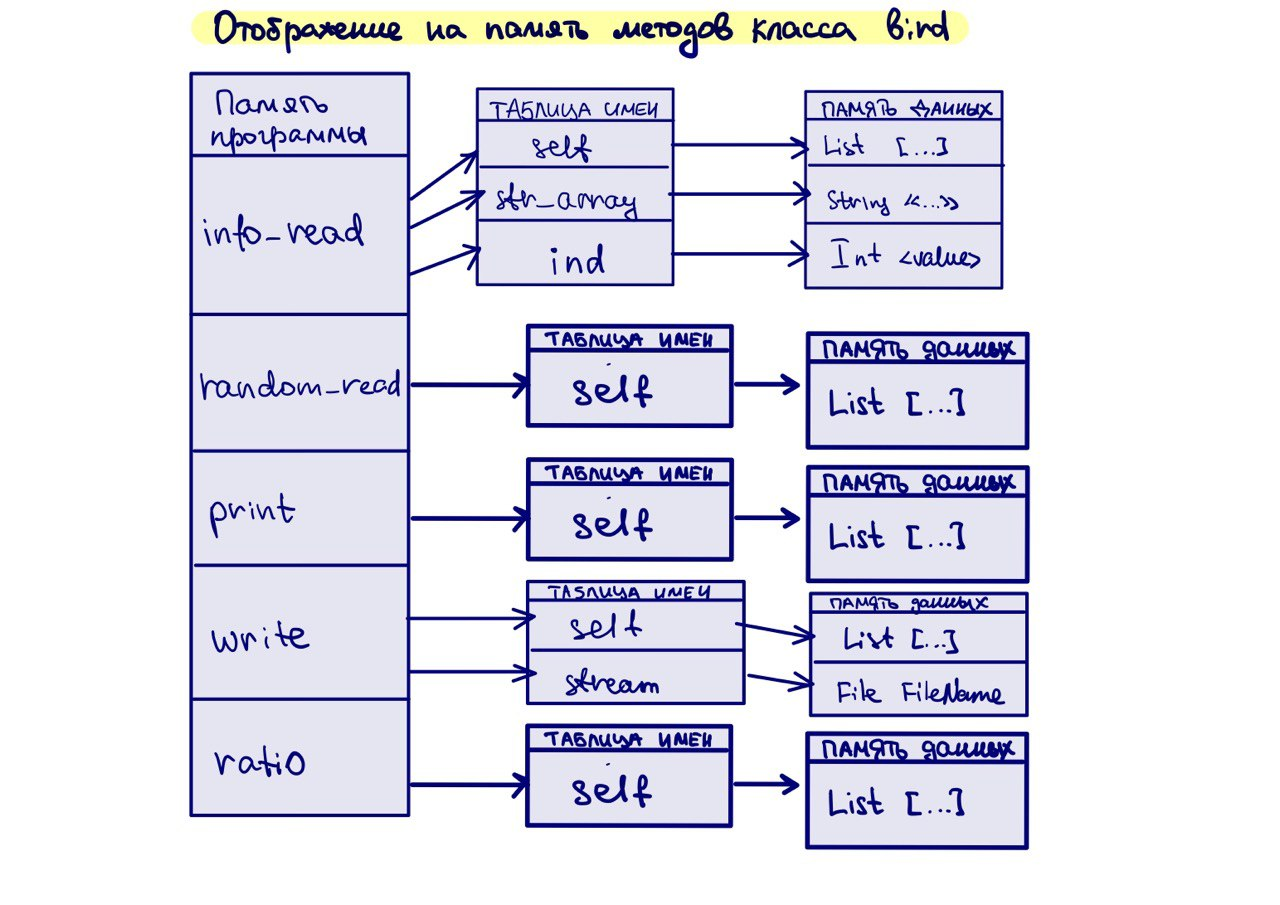


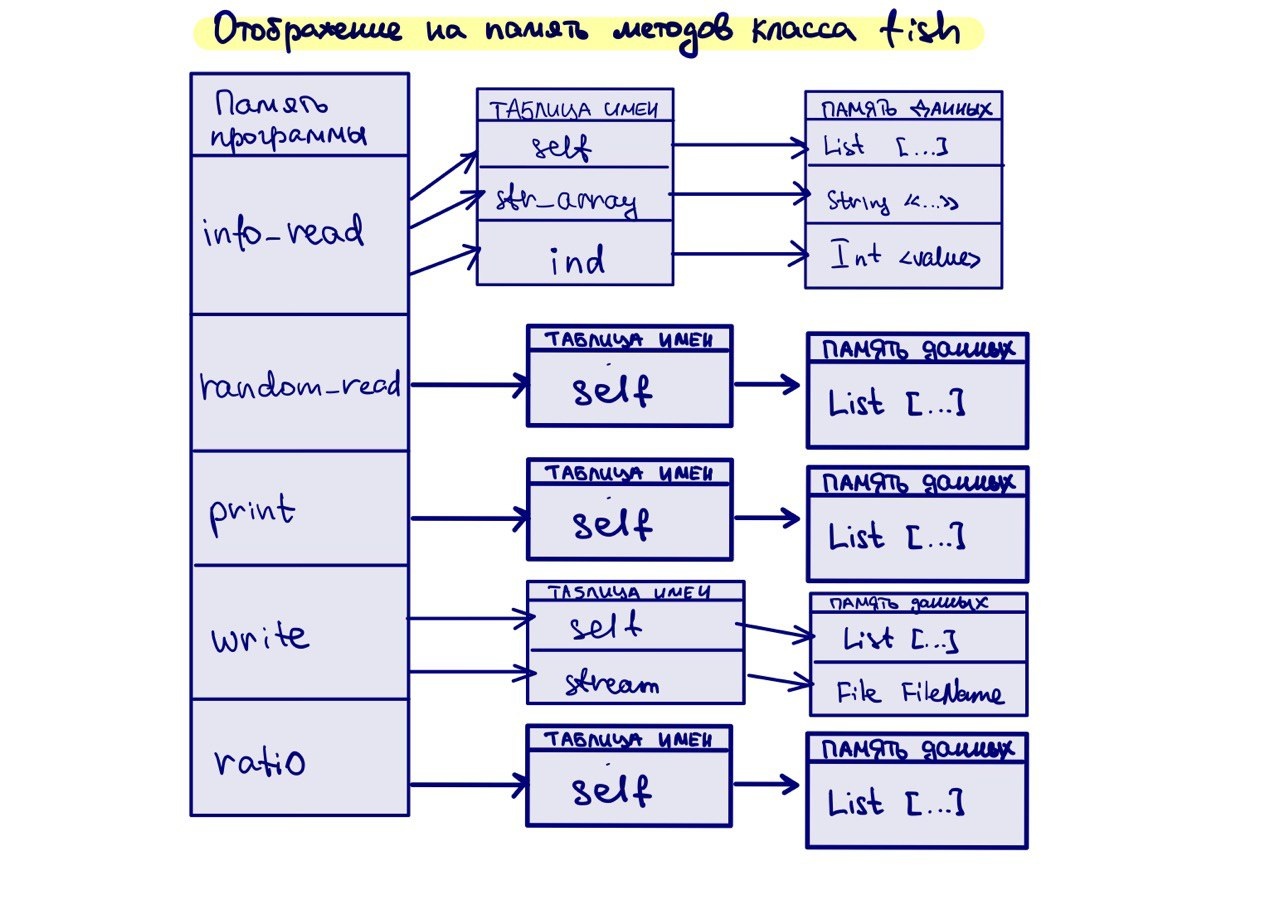












1. **Основные характеристики программы**

1) число интерфейсных модулей (заголовочных файлов) – 0

2) число модулей реализации (фалов с определением программных объектов) – 7

3) общий размер исходных текстов – 1+3+2+3+3+2+3= 17 Кб

4) время выполнения программы для различных тестовых наборов данных:

1 тест:

elapsed time: 0.00106 seconds

2 тест:

elapsed time: 0.00291 seconds

3 тест:

elapsed time: 0.006 seconds

4 тест:

elapsed time: 0.00192 seconds

5 тест:

elapsed time: 0.00394 seconds

1. **Ввод данных**

**3 Vobla 54 2**

**1 Rabbit 23 2  
2 Snipe 25 1**

Первая переменная - вид животного (1 – зверь, 2 – птица, 3 – рыба), вторая переменная – название животного, третья переменная – за его вес в граммах, четвертая переменная – за дополнительные параметры.

Параметры для рыбы – это место обитания:

1- River, 2 - sea, 3 - lake

Параметр для птицы – переменная, которая отвечает за то, является ли птица перелетной

0 – нет

1 – да

Параметры для зверя – тип животного:

1. - Predators, 2 - herbivores, 3 - insectivores

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.