**Лабораторная работа №8**

*Задача 1.*

Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

* наименование радиостанции (строка до 60 символов);
* частота вещания (вещественное число).

Наименование радиостанции может иметь несколько слов, разделенных пробелами. Вычислить количество радиостанций, вещающих в заданном диапазоне частот.

Для реализации решения поставленной задачи используется язык программирования Python.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | k=0  s=0  count = 0  def search(ch,name):  y = int(ch)  global count  if ( (y > k) & (y < s) ):  print(y)  count=count+1  print("Колличество радиостанций, частоты которых входят в диапазон:",count)  class RadStation:  def \_\_init\_\_(self, one, two):  self.name=one  self.frequency=two  def show(self):  namestr=self.name  print('\nНазвание радиостанции',self.name, "Частота:", self.frequency)  def poinr(selfs):  str2=selfs.name  ch =selfs.frequency  xad=search(ch,str2)  print('Введите диапазаон:')  k=int(input())  s=int(input())  well = RadStation("Radio Hill",4)  well.show()  well.poinr()  sqtr =RadStation("Станция 2",5)  sqtr.show()  sqtr.poinr()  yt=RadStation("Рыба", 7)  yt.show()  yt.poinr()  tr=RadStation("Мы тут",78)  tr.show()  tr.poinr() |

## Оценка характеристик программы

Метрика размера класса CS определяется на основание следующих показателей:

* общее количество операций;
* количество свойств, а значение рассчитывается по формуле

где С∑ - количество инкапсулированных классом методов; 𝑆∑ - количество инкапсулируемых классом свойств (полей классов).

1. Для класса ***RadStaition*** число инкапсулированных методов С∑ = 3, количество инкапсулированных полей 𝑆∑ = 0.

Количество посылаемых сообщений равно 2. Таким образом, для класса ***RadStaition*** 𝐴𝑂𝑆 = 2.

Полученные значения метрик классов не превышают значения 20, следовательно, сложность классов не превышает требуемый уровень и разбиение программы на дополнительные модули не требуется.

Количество операций, переопределяемых подклассом 𝑁𝑂𝑂, количество операций, добавленных подклассом 𝑁𝑂А, индекс специализации 𝑆𝐼 для данного решения не определяются, так как в исходном коде программы не реализован принцип наследования.

Из полученных результатов следует, что наиболее сложным является класс ***RadStaition.***

Определим сложность операции ОС, которую будем вычислять по количеству строк в коде метода. При вычислении воспользуемся весомыми коэффициентами, предложенными Лоренцом и Киддом.

def \_\_init\_\_(self, one, two):

|  |  |  |
| --- | --- | --- |
| Действие | Вес | Количество строк |
| Определение (описание) переменной-параметра | 0,3 |  |
| Определение (описание) временной переменной | 0,5 |  |
| Присваивание значений | 0,5 | 2 |
| Вложенное выражение | 0,5 |  |
| Сообщение без параметров | 1 |  |
| Арифметическая операция | 2 |  |
| Сообщение с параметрами | 3 |  |
| Вызов стандартной функции интерфейса (API) | 5 | 1 |
| Вызов пользовательской функции (простой вызов) | 7 | 2 |

Расчет сложности операции ОС дает следующий результат: ОС = 0,5 ∗ 2 +5+14= 20.

Метод def poinr(selfs):

|  |  |  |
| --- | --- | --- |
| Действие | Вес | Количество строк |
| Определение (описание) переменной-параметра | 0,3 | 0 |
| Определение (описание) временной переменной | 0,5 | 0 |
| Присваивание значений | 0,5 | 2 |
| Вложенное выражение | 0,5 | 0 |
| Сообщение без параметров | 1 | 0 |
| Арифметическая операция | 2 | 0 |
| Сообщение с параметрами | 3 | 0 |
| Вызов стандартной функции интерфейса (API) | 5 | 0 |
| Вызов пользовательской функции (простой вызов) | 7 | 0 |

Расчет сложности операции ОС дает следующий результат: ОС = 0,5 ∗ 2 = 1.

Метод def search(str):

|  |  |  |
| --- | --- | --- |
| Действие | Вес | Количество строк |
| Определение (описание) переменной-параметра | 0,3 | 0 |
| Определение (описание) временной переменной | 0,5 | 0 |
| Присваивание значений | 0,5 | 3 |
| Вложенное выражение | 0,5 | 0 |
| Сообщение без параметров | 1 | 0 |
| Арифметическая операция | 2 | 0 |
| Сообщение с параметрами | 3 | 0 |
| Вызов стандартной функции интерфейса (API) | 5 | 0 |
| Вызов пользовательской функции (простой вызов) | 7 | 0 |

Расчет сложности операции ОС дает следующий результат: ОС = 0,5 ∗ 3 = 1,5.

Метод def show(self):

|  |  |  |
| --- | --- | --- |
| Действие | Вес | Количество строк |
| Определение (описание) переменной-параметра | 0,3 | 0 |
| Определение (описание) временной переменной | 0,5 | 0 |
| Присваивание значений | 0,5 | 2 |
| Вложенное выражение | 0,5 | 0 |
| Сообщение без параметров | 1 | 0 |
| Арифметическая операция | 2 | 0 |
| Сообщение с параметрами | 3 | 0 |
| Вызов стандартной функции интерфейса (API) | 5 | 0 |
| Вызов пользовательской функции (простой вызов) | 7 | 0 |

Расчет сложности операции ОС дает следующий результат: ОС = 0,5 ∗ 2= 1.

Определим среднее количество параметров на операцию 𝐴𝑁𝑃 (𝐴𝑣𝑒𝑟𝑎𝑔𝑒 𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑃𝑎𝑟𝑎𝑚𝑒𝑡𝑒𝑟𝑠 𝑝𝑒𝑟 𝑜𝑝𝑒𝑟𝑎𝑡𝑖𝑜𝑛) , которое рассчитывается по следующему соотношению:

* Для класса ***RadStaition t***

Определим количество сценариев 𝑁𝑆𝑆 (𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑆𝑐𝑒𝑛𝑎𝑟𝑖𝑜 𝑆𝑐𝑟𝑖𝑝𝑡𝑠), опираясь на количество методов в классе.

Количество сценариев для исследуемой программы невелико, что объясняет сравнительно небольшие размеры программы.

Количество ключевых классов 𝑁𝐾𝐶 (𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝐾𝑒𝑦 𝐶𝑙𝑎𝑠𝑠𝑒𝑠) в рассматриваемой программе равно трем. Все классы, определенные в программе, можно назвать ключевыми, так как они напрямую связаны с проблемной областью поставленной задачи. Таким образом, 𝑁𝐾𝐶 = 1, т. е. количество ключевых классов от общего количества классов программы составляет 100%.

В рассматриваемом решении можно выделить одну подсистему:

* класс, определяющий ***RadStaition*** как объект для вычисления количества радиостанций в заданном диапазоне частот;

Таким образом, количество подсистем 𝑁𝑆𝑈𝐵 (𝑁𝑢𝑚𝑏𝑒𝑟 𝑜𝑓 𝑠𝑢𝑏𝑠𝑦𝑠𝑡𝑒𝑚) = 1. Количество подсистем программы обеспечивает достаточную управляемость проекта и невысокую трудоемкость.