Лабораторная работа №7

*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("Тут мы",78)  tr.show()  tr.poinr() |

*Оценка характеристик программы.*  
Проанализируем текст программы для оценки ее качества с помощью метрик Мартина, которые позволяют оценить меру сложности объектно-ориентированной программы на основе анализа организационной структуры классов программы и их связности, так как все классы работают во взаимодействии.

Исходный код программы включает один класс, который можно отнести к одной категории, предназначенной для решения задачи, целью которой является подсчет количества радиостанций в заданном диапазоне частот:

* *RadStaition* - базовый класс, определяющий понятие *радиостанция*;

Класс *RadStaition* в своем составе содержит три метода:

* ***def \_\_init\_\_(self, one, two)***
* ***def show(self)***
* ***def poinr(selfs)***

𝑊𝑀𝐶 (*RadStaition*) = 3

Методы состоят из:

* ***def \_\_init\_\_(self, one, two)*** - 2 строк
* ***def show(self) -*** 2 строк
* ***def poinr(selfs) -*** 3 строк

Сложность класса ***RadStaition*** определяется следующим образом:

(***RadStaition***) = 𝐿[0] + 𝐿[1] + 𝐿[2] = 2 + 2 + 3 = 7.

Cамым сложным является класс ***RadStaition***, так как этот класс содержит наибольшее количество методов.

Определим связность между классами объектов СВО, которая численно определяется по количеству классов, с которыми связан анализируемый класс. Связь класса с другими классами определяется количеством обращений к вызову методов в других классах.

1. В классе ***RadStaition*** не наблюдаются обращения к методам других классов.Следовательно,

СВО (***RadStaition***) = 0.

1. В исходном коде класса **Main** есть обращение к методу класса ***RadStaition* .** Следовательно

СВО (**Main**) = 1.

Наиболее сложным по связности является основной класс **Main**, хотя остальные классы имеют достаточно низкую связность объектов.

Определим количество откликов на класс RFC, которое определяется числом методов, которые могут быть вызваны в рассматриваемом классе. Значение метрики RFC определяется следующим соотношением:

𝑅𝐹𝐶 = 𝑅 + 𝑀,

где R - это методы вызываемые методами рассматриваемого класса,   
М - общее количество методов класса.

𝑅𝐹𝐶(**Staition**) = 0 + 3 = 3;

𝑅𝐹𝐶(**Main**) = 1 + 3 = 4;

Определим значение метрики отсутствия сцепления в методах LCOM. Числовое значение метрики определяется разностью между количеством пар методов, которые не используют одни и те же переменные в классе, и количеством пар методов, которые используют одни и те же переменные класса.

Класс ***RadStaition***. Методы, определенные в этом классе, попарно обращаются к одним и тем же переменным, поэтому 𝐿𝐶𝑂𝑀 = 3 - 0 = 3, следовательно, считаем значение LCOM = 3.

Анализируя значения метрик LCOM всех классов программы, можно отметить, что уровень связности методов в классах достаточно высок, что позволяет в полной мере обеспечить инкапсуляцию классов.