

ОЦЕНКА ХАРАКТЕРИСТИК ПРОГРАММ НА ОСНОВЕ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ МЕТРИК

Метрики Мартина

Теоретические сведения

Класс представляет собой абстракцию, которая может быть размещена на различных уровнях детализации, причем разными способами (например, в виде списка операций, последовательности состояний или последовательности взаимодействий). В связи с этим для объектно-ориентированных метрик более целесообразно представление абстракций в терминологии измерений класса. К примерам такого представления можно отнести количество экземпляров класса в приложении, количество родовых классов в приложении, отношение количества родовых классов к количеству неродовых классов.

В 1995 г. американский консультант по программному обеспечению Роберт Сесил Мартин (Robert Cecil Martin) предложил пять основных принципов дизайна классов в объектно-ориентированном проектировании (так называемая методология *SOLID*) и разработал комплекс метрик, получивших наименование метрик Мартина.

В программах класс редко может быть повторно использован изолированно от других классов. Почти всегда класс имеет группу классов, с которыми он работает во взаимодействии и от которых его достаточно трудно отделить. Для повторного использования таких классов необходимо заново использовать всю группу классов. Связность такой группы классов (она называется *категорией классов*) достаточно высока, и для ее существования должны быть соблюдены следующие условия:

- классы в пределах категории закрыты от любых попыток изменения отдельных экземпляров. Это означает, что если одному классу необходимо измениться, то весьма вероятно изменение всех классов в этой категории. Если любой из классов открыт для

некоторого вида изменений, они все открыты для этого вида изменений;

- классы в категории повторно используются только вместе. Они настолько взаимозависимы, что не могут быть отделены друг от друга. Поэтому если предпринимается попытка повторного использования одного класса в категории, то все другие классы этой категории также повторно используются вместе с таким классом;
- классы в категории обеспечивают некоторую общую функцию или достигают некоторую общую цель.

Ответственность, независимость и стабильность категории, по мнению Мартина, могут быть измерены путем подсчета некоторых зависимостей, взаимодействующих с этой категорией. Поскольку данные зависимости не являются основными, опустим их рассмотрение.

Мартин предложил следующие метрики для оценки характеристик программы при объектно-ориентированном программировании:

- Ca (центростремительное сцепление) - метрика, определяющая количество классов вне конкретной категории, которые зависят от классов внутри нее;
- Ce (центробежное сцепление) - метрика, оценивающая количество классов внутри конкретной категории, которые зависят от классов вне нее;
- I (нестабильность) - расчетная метрика, определяемая в соответствии с выражением

$$I = \frac{Ce}{Ca + Ce}$$

Данная метрика имеет диапазон значений $[0, 1]$: значение метрики $I = 0$ означает максимально стабильную категорию, а $I = 1$ указывает максимально нестабильную категорию.

Дополнительно к указанным метрикам можно определять меру A

(*абстрактность*), которая позволяет оценить абстрактность категории (если категория абстрактна, то она является достаточно гибкой и может быть легко расширена):

$$A = \frac{nA}{nAll},$$

где nA - количество абстрактных классов в категории; $nAll$ - общее количество классов в категории.

Значения метрики абстрактности расположены в диапазоне $[0, 1]$: при нулевом значении категория полностью конкретна, а при единичном значении является полностью абстрактной.

На основе приведенных метрик Мартина можно построить график, отражающий зависимость между абстрактностью и нестабильностью. Если построить прямую, задаваемую формулой $I + A = 1$, то категории, расположенные на этой прямой, будут иметь наилучшую сбалансированность между абстрактностью и нестабильностью. Эта прямая называется *главной последовательностью*.

Получив главную последовательность, можно ввести еще две метрики:

- нормализованное расстояние до главной последовательности:

$$D = \left| \frac{A + I - 1}{\sqrt{2}} \right|$$

- расстояние до главной последовательности:

$$D_n = |A + I - 2|$$

Практически для любых категорий классов справедливым является следующее утверждение: чем ближе они находятся к главной последовательности, тем лучше для обеспечения качества программного средства.

Примеры

Задача «Платеж за электроэнергию»

Необходимо определить класс, описывающий платеж за электроэнергию.

В рамках класса следует предусмотреть следующие поля:

- фамилия плательщика;
- потребление электроэнергии за оплачиваемый месяц;
- нормативное среднеемесячное потребление;
- тариф (стоимость одного киловатт-часа).

Рекомендуется применять следующие методы:

- вычисление суммы оплаты;
- формирование сводной информации по одному платежу (вид платежа, фамилия, сумма, потребление).

Платеж может выполняться по показаниям счетчика или по нормативно установленному среднеемесячному потреблению. В процессе эксплуатации программы тариф и нормативно установленное среднеемесячное потребление не изменяются. Для создания конкретного платежа предусмотреть соответствующий конструктор.

Все платежи должны сохраняться в архиве. Запросы по ведению архива выполняются статическими методами класса «Запрос»:

- занесение платежей в архив. Данные платежа вводятся с клавиатуры. Ввод отрицательного показания счетчика означает оплату по нормативно установленному среднеемесячному потреблению;
- вывод сводной информации из архива.

Архив моделируется массивом объектов.

В основном классе «Платежи» следует сформировать архив платежей. По данным архива предусмотреть выдачу сводной информации о платежах.

При решении задачи необходимо разработать исходный код программы, а также определить оценки характеристик программы на основе объектно-ориентированных метрик Мартина.

Реализация программы

Текст программы на языке С# для реализации возможного алгоритма решения поставленной задачи представлен на рис. 1.

Номера строк	Строки программы
1	using System;
2	using System.Collections.Generic;
3	using System.Text;
4	namespace EX1
5	{
6	class Электро
7	{
8	private static double тариф = 1.84;
9	private static int потреблениеСреднее = 300;
10	private string фамилия;
11	private int потребление;
12	public Электро(string фамилия)
13	{
14	this.фамилия = фамилия;
15	потребление = потреблениеСреднее;
16	}
17	public Электро(string фамилия, int текущее, int предыдущее)
18	{
19	this.фамилия = фамилия;
20	потребление = текущее - предыдущее;
21	}
22	public double Сумма()
23	{
24	return потребление * тариф;
25	}
26	public string Инфо()
27	{
28	return string.Format("{0,-20} {1,-20} {2,10:f2} {3,10:d6}",
29	"Электроэнергия", фамилия, Сумма(), потребление);
30	}
31	}
32	class Запрос
33	{
34	public static void Заполнить(Электро[] платеж)
35	{
36	string фам;
37	int счПред=0, счТек=0;
38	Console.Clear();
39	for (int i = 0; i < платеж.Length; i++)
40	{
41	Console.Write("Платеж " + i + ": Фамилия -> ");
42	фам = Console.ReadLine();
43	Console.Write("Платеж " + i + ": Текущее значение счетчика -> ");
44	счТек = int.Parse(Console.ReadLine());
45	if (счТек > 0)
46	{
47	Console.Write("Платеж " + i + ": Предыдущее значение счетчика -> ");


```

48     счПред = int.Parse(Console.ReadLine());
49     }
50     if(счТек <= 0)
51         платеж[i] = new Электро(фам);
52     else
53         платеж[i] = new Электро(фам, счТек, счПред);
54     }
55     }
56     public static void Вывести(Электро[] платеж)
57     {
58         for (int i = 0; i < платеж.Length; i++)
59             Console.WriteLine(платеж[i].Инфо());
60     }
61     }
62     class Платежи
63     {
64     static void Main(string[] args)
65     {
66         ConsoleKeyInfo rep;
67         Электро[] плэ;
68         int кпэ;
69         do
70         {
71             Console.Clear();
72             Console.Write("Количество платежей за электроэнергию: ");
73             кпэ = int.Parse(Console.ReadLine());
74             плэ = new Электро[кпэ];
75             Запрос.Заполнить(плэ);
76             Запрос.Вывести(плэ);
77             Console.WriteLine("Для выхода нажмите ESC");
78             rep = Console.ReadKey(true);
79             }while(rep.Key!= ConsoleKey.Escape);
80         }
81     }
82     }

```

Рис. 4.1. Пример реализации программы «Платеж за электроэнергию»

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 детализации организационной структуры классов программы и их связности, так как все классы работают во взаимодействии.

Исходный код программы включает три класса, которые можно отнести к одной категории, предназначенной для решения задачи, целью которой является расчет сумм платежей за электроэнергию.

К ним относятся следующие классы (см. рис. 1):

- **class** Электро (строка 6) - класс, определяющий экземпляр объекта платежа за электроэнергию;
- **class** Запрос (строка 32) - класс, содержащий методы обработки информации об объектах типа Электро;
- **class** Платежи (строка 62) - класс, в котором содержится основной метод обработки информации массивов объектов типа Электро с помощью методов класса Запрос.

Кроме указанных классов в программе используются обращения к библиотеке встроенных классов среды программирования, которые не входят в рассматриваемую категорию классов. Необходимо отметить, что классы, не входящие в рассматриваемую категорию, никак не зависят от ее классов. Следовательно, центростремительное сцепление классов по теории Мартина $S_f = 0$. Нулевое значение метрики определяет полную независимость классов других категорий, не связанных с расчетом платежей за электроэнергию.

В классе Электро используется метод класса **string** (строка 25), в классах Запрос и Платежи используются методы класса **Console** (строки 35, 38, 39, 40, 42, 68, 69, 70). Таким образом, все три класса рассматриваемой категории имеют зависимость от встроенных классов, которые не являются ее составными частями. Поэтому центробежное сцепление классов по теории Мартина $S_u = 3$.

Расчетная метрика нестабильности I определяется следующим образом:

$$I = \frac{C_e}{C_e + C_a} = \frac{3}{3 + 0} = 1$$

Исходя из полученного значения ($I = 1$) следует, что категория классов анализируемой программы является максимально нестабильной, так как зависимость от классов других категорий носит преобладающий характер.

Определим меру абстрактности A рассматриваемой категории классов. Код программы не содержит ни одного абстрактного класса, следовательно, $nA = 0$. Общее количество классов в рассматриваемой категории составляет 3, следовательно, $nAll = 3$.

$$A = \frac{nA}{nAll} = \frac{0}{3} = 0.$$

Исходя из значения метрики $A = 0$ можно сделать вывод, что рассматриваемая категория классов является полностью конкретной.

Расстояние до главной последовательности определяется уравнением $I + A = 1$. В соответствии с теорией Мартина расстояние до главной последовательности определяется формулой

$$D = \left| \frac{A + I - 1}{\sqrt{2}} \right| = \left| \frac{0 + 1 - 1}{\sqrt{2}} \right| = 0$$

Категория расположена на прямой главной последовательности, что свидетельствует о наилучшей сбалансированности между абстрактностью и нестабильностью. Нормализованное расстояние до главной последовательности равно:

$$D_n = |A + I - 2| = |0 + 1 - 2| = 1$$

В целом качество разработанной программы можно считать достаточно высоким, так как рассматриваемая категория классов исходного кода находится в пределах нормализованного расстояния до главной последовательности.

Задача «Геометрия окружности и прямоугольника»

В предметной области «Геометрия» определены информационные объекты, описывающие понятия «Точка», «Окружность» «Прямоугольник» в виде классов. Классы «Окружность» и «Прямоугольник» являются наследниками класса «Точка». Каждый из классов имеет метод вывода параметров объекта. Следует иметь в виду, что метод вывода в базовом классе не виртуальный и не абстрактный. Методы вывода в наследниках переопределены. Вызываемый метод будет определяться типом ссылки.

Базовый класс «Точка» описан следующим образом:

- закрытые поля:
 - координата по оси X (вещественная);
 - координата по оси Y (вещественная);
 - метод: вывод координат точки.
- Класс «Окружность» задан следующими элементами:
- закрытые поля:
 - координаты центра - унаследованные поля;
 - радиус (вещественный);
 - метод: вывод координат центра и длины окружности.

Класс «Прямоугольник» задается так:

- закрытые поля:
 - координаты левого верхнего угла - унаследованные поля;
 - основание (вещественное);
 - высота (вещественная);
- метод: вывод координат левого верхнего угла и площади.

В основном классе необходимо сформировать список геометрических объектов разных типов в виде массива и вывести параметры объектов на экран. Разработать исходный код программы, по которому определить оценки характеристик программы на основе объектно-ориентированных метрик Мартина.

Реализация программы

Текст программы на языке C# для реализации возможного алгоритма решения поставленной задачи представлен на рис. 2.

Номера строк	Строки программы
1	namespace EX1_1
2	{
3	class Точка
4	{
5	protected double x, y;
6	public Точка(double x, double y)
7	{
8	this.x = x; this.y = y;
9	}
10	public void Показать()
11	{
12	Console.WriteLine("Точка ({0:f2}:{1:f2})", x, y);
13	}
14	}
15	class Окружность: Точка
16	{
17	private double r;
18	public Окружность(double x, double y, double r): base(x, y)
19	{
20	this.r = r;
21	}
22	public new void Показать()
23	{
24	Console.WriteLine("Центр({0:f2}:{1:f2}) Длина={2:f2}", x, y, 2*Math.PI*r);
25	}
26	}
27	
28	class Прямоугольник: Точка
29	{
30	private double a, h;


```

31 public Прямоугольник(double x, double y, double a, double h): base(x, y)
32 {
33     this.a = a; this.h = h;
34 }
35 public new void Показать()
36 {
37     Console.WriteLine("Угол({0:f2}:{1:f2}) Площадь={2:f2}", x, y, a*h);
38 }
39 }
40 class Program
41 {
42     static void Main(string[] args)
43     {
44         Точка[] список = new Точка[6];
45         список[0] = new Точка(-1.0, -1.0);
46         список[1] = new Окружность(1.0, 1.0, 1.0);
47         список[2] = new Прямоугольник(10.0, 10.0, 10.0, 10.0);
48         список[3] = new Точка(-2.0, -2.0);
49         список[4] = new Окружность(3.0, 3.0, 3.0);
50         список[5] = new Прямоугольник(20.0, 20.0, 0.5, 0.5);
51         for (int i = 0; i < список.Length; i++) список[i].Показать();
52         Console.WriteLine();
53         for (int i = 0; i < список.Length; i++)
54         {
55             if (список[i] is Точка)
56                 ((Точка)список[i]).Показать();
57             else
58                 if (список[i] is Окружность)
59                     ((Окружность)список[i]).Показать();
60                 else
61                     if (список[i] is Прямоугольник)
62                         ((Прямоугольник)список[i]).Показать();
63                 }
64             Console.WriteLine();
65             for (int i = 0; i < список.Length; i++)
66                 switch (список[i].GetType().Name)
67                 {
68                     case "Точка": ((Точка)список[i]).Показать(); break;
69                     case "Окружность": ((Окружность)список[i]).Показать(); break;
70                     case "Прямоугольник": ((Прямоугольник)список[i]).Показать(); break;
71                 }
72             }
73         }
74     }

```

Рис. 4.2. Пример реализации программы «Геометрия»

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 четыре класса, которые можно отнести к одной категории, предназначенной для решения задачи, целью которой является определение геометрических фигур на основании исходных координат точек, определяющих местоположение этих фигур. К ним относятся следующие классы (см. рис. 4.2):

- ***class* Точка** - базовый класс, определяющий понятие *точка*, расположенная на координатной плоскости;
- ***class* Окружность** - производный класс от *class Точка*, определяющий понятие *окружность* с центром в точке, расположенной на координатной плоскости;
- ***class* Прямоугольник** - производный класс от *class Точка*, определяющий понятие *прямоугольник*, расположение которого на плоскости определяется координатами точки верхнего левого его угла;
- ***class Program*** - класс, в котором содержится основной метод обработки массивов объектов типа *Точка*, *Окружность* и *Прямоугольник*.

Кроме указанных классов в программе используются обращения к библиотеке встроенных классов среды программирования, которые не входят в рассматриваемую категорию классов.

Необходимо отметить, что классы, не входящие в рассматриваемую категорию, никак не зависят от ее классов. Следовательно, центростремительное сцепление классов по теории Мартина $C_a = 0$. Нулевое значение метрики определяет полную независимость классов других категорий, не связанных с обработкой информации по определению геометрических фигур.

В классе *Точка* осуществляется вызов метода из класса *Console* (строка 12). В классе *Окружность* осуществляется вызов методов из классов *Console* и *Math* (строка 24). В классе *Прямоугольник* осуществляется вызов метода из класса *Console* (строка 37). Класс *Program* содержит обращения к методам класса *Console* (строки 52, 64). Таким образом, все четыре класса программы имеют связь с классами, не входящими в рассматриваемую категорию, и зависят от них. Следовательно, центробежное сцепление классов по теории Мартина $Ce = 4$.

Расчетная метрика нестабильности I определяется следующим образом:

$$I = \frac{Ce}{Ce + Ca} = \frac{4}{4 + 0} = 1$$

Исходя из полученного значения ($I = 1$) следует, что категория классов анализируемой программы является максимально нестабильной, так как зависимость от классов других категорий носит преобладающий характер.

Определим меру абстрактности A рассматриваемой категории классов. Код программы не содержит ни одного абстрактного класса, следовательно, $pA = 0$. Общее количество классов в рассматриваемой категории составляет 4, следовательно, $nAll = 4$.

$$A = \frac{nA}{nAll} = \frac{0}{4} = 0$$

Исходя из значения метрики $A = 0$ можно сделать вывод, что рассматриваемая категория классов является полностью конкретной.

Расстояние до главной последовательности определяется уравнением $I + A = 1$. В соответствии с теорией Мартина расстояние до главной последовательности определяется формулой

$$D = \left| \frac{A + I - 1}{\sqrt{2}} \right| = \left| \frac{0 + 1 - 1}{\sqrt{2}} \right| = 0.$$

Категория расположена на прямой главной последовательности, что свидетельствует о наилучшей сбалансированности между абстрактностью и нестабильностью. Нормализованное расстояние до главной последовательности равно:

$$D_n = |A + I - 2| = |0 + 1 - 2| = 1$$

В целом качество разработанной программы можно считать достаточно высоким, так как рассматриваемая категория классов исходного кода находится в пределах нормализованного расстояния до главной последовательности.

Задачи для самостоятельного решения

В предлагаемых задачах необходимо разработать программу, реализующую решение по приведенному условию, а затем оценить характеристики разработанной программы на основе применения объектно-ориентированных метрик Мартина.

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.

Задача 2. Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

- наименование радиостанции (строка до 60 символов);
- частота вещания (длинное целое число).

Наименование радиостанции может иметь несколько слов, разделенных пробелами. В таблице радиостанций изменить название радиостанции, вещающей на заданной частоте.

Задача 3. Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

- наименование радиостанции (строка до 60 символов);
- частота вещания (длинное целое число).

Наименование радиостанции может иметь несколько слов, разделенных пробелами. В таблице радиостанций изменить частоту вещания радиостанции с заданным названием.

Задача 4. 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, имеющих в своем названии сочетание букв «музыка».

Задача 5. 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, имеющих в своем названии от 2 до 4 слов.

Задача 6. Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

- наименование радиостанции (строка до 60 символов);
- частота вещания (длинное целое число).

Наименование радиостанции может иметь несколько сл'бв, разделенных пробелами. Составляется заявка на распределение частот вещания. Проверить таблицу радиостанций на корректность. Сформировать и вывести на экран наименования радиостанций, подавших заявки на вещание на совпадающих частотах. При отсутствии конфликтов выдать сообщение «Норма».

Задача 7. Определить понятие «Точка». Состояние объекта определяется следующими полями:

- координата по оси абсцисс (вещественное число);
- координата по оси ординат (вещественное число).

Базируясь на понятии «Точка», определить понятие «Квадрат».

Квадрат определить через точку, соответствующую его левому верхнему углу, и размер стороны квадрата. Проверить попадание точки в квадрат.

Задача 8. Определить понятие «Книга». Состояние объекта определяется следующими полями:

- единый регистрационный номер (длинное целое число);
- автор (строка до 20 символов);
- год издания (целое число);
- количество экземпляров (целое число).

Вычислить количество экземпляров книг заданного автора, выпущенных в период с 1986 по 2001 г.

Задача 9. Определить понятие «Книга». Состояние объекта определяется следующими полями:

- единый регистрационный номер (длинное целое число);
- автор (строка до 20 символов);
- год издания (целое число);
- количество страниц (целое число).

Вычислить средний объем книги в страницах среди книг, выпущенных за заданный период времени.

Задача 10. Определить понятие «Точка». Состояние объекта определяется следующими полями:

- координата по оси абсцисс (вещественное число);
- координата по оси ординат (вещественное число).

Базируясь на понятии «Точка», определить понятие «Квадрат».

Квадрат определить через точку, соответствующую его левому верхнему углу и размер стороны квадрата.

Изменить размер стороны квадрата на заданную величину DR и вычислить площадь квадрата после изменения его размера.

Задача 11. Определить понятие «Книга». Состояние объекта определяется следующими полями:

- единый регистрационный номер (длинное целое число);
- наименование (строка до 50 символов);
- автор (строка до 20 символов);

год издания (целое число);

количество экземпляров (целое число).

Наименование книги может иметь несколько слов, разделенных пробелами. Увеличить количество экземпляров книги с заданным единым регистрационным номером на 100 единиц.

Задача 12. Определить понятие «Книга». Состояние объекта определяется следующими полями:

единый регистрационный номер (длинное целое число);

наименование (строка до 50 символов);

автор (строка до 20 символов);

год издания (целое число);

количество экземпляров (целое число).

Наименование книги может иметь несколько слов, разделенных пробелами. Изменить для книги с заданным регистрационным номером фамилию автора на значение «Иванов».

Задача 13. Определить понятие «Точка». Состояние объекта определяется следующими полями:

координата по оси абсцисс (вещественное число);

координата по оси ординат (вещественное число).

Базируясь на понятии «Точка», определить понятие «Прямоугольник». Прямоугольник определить через точки, соответствующие его левому верхнему и правому нижнему углам. Вычислить площадь и периметр прямоугольника.

Задача 14. Определить понятие «Точка». Состояние объекта определяется следующими полями:

координата по оси абсцисс (вещественное число);

координата по оси ординат (вещественное число).

Базируясь на понятии «Точка», определить понятие «Прямоугольник». Прямоугольник определить через точки, соответствующие его левому верхнему и правому нижнему углам. Сдвинуть правый нижний угол прямоугольника по обеим осям на заданное расстояние *DELTA* и вычислить периметр

прямоугольника после сдвига угла.

Задача 15. Определить понятие «Автомобиль». Состояние объекта определяется следующими полями:

регистрационный номер (строка из 6 символов);

код региона России (целое число);

год выпуска (целое число);

фамилия владельца (строка до 20 символов).

В таблице регистрации автомобилей по заданному регистрационному номеру и коду региона изменить фамилию владельца на значение «Петровский».

Задача 16. Определить понятие «Автомобиль». Состояние объекта определяется следующими полями:

регистрационный номер (строка из 6 символов);

код региона России (целое число);

год выпуска (целое число). *

В таблице регистрации автомобилей удалить данные обо всех автомобилях, выпущенных до заданного года. Вывести модифицированную таблицу.

Задача 17. Определить понятие «Человек». Состояние объекта определяется следующими полями:

номер паспорта (семизначное целое число);

фамилия (строка до 20 символов).

Базируясь на указанном понятии, определить понятие «Проживающий», дополнительно определив понятие «Человек» полями «Наименование отеля» (строка до 15 символов) и «Количество суток проживания» (целое число). Вычислить сумму, полученную в качестве оплаты за проживание в отеле «Приморский» от всех проживающих.

Задача 18. Определить понятие «Автомобиль». Состояние объекта определяется следующими полями:

регистрационный номер (строка из 6 символов);

код региона России (целое число);

год выпуска (целое число);

фамилия владельца (строка до 20 символов).

По таблице регистрации автомобилей вычислить количество автомобилей с регистрационным номером «А495ВО» в заданных регионах России.

Задача 19. Определить понятие «Автомобиль». Состояние объекта определяется следующими полями:

регистрационный номер (строка из 6 символов);

код региона России (целое число);

год выпуска (целое число);

фамилия владельца (строка до 20 символов).

По таблице регистрации автомобилей вывести однофамильцев среди владельцев автомобилей, выпущенных в заданном году.

Задача 20. Определить понятие «Книга». Состояние объекта определяется следующими полями:

- единый регистрационный номер (длинное целое число);
- автор (строка до 20 символов);
- год издания (целое число);
- количество страниц (целое число).

Вычислить средний объем книги в страницах среди книг, выпущенных за заданный период времени.

Задача 21. Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

- наименование радиостанции (строка до 60 символов);
- частота вещания (длинное целое число).

Наименование радиостанции может иметь несколько слов, разделенных пробелами. В таблице радиостанций изменить название радиостанции, вещающей на заданной частоте.

Задача 22. Определить понятие «Радиостанция». Состояние объекта определяется следующими полями:

- наименование радиостанции (строка до 60 символов);
- частота вещания (длинное целое число).

Наименование радиостанции может иметь несколько слов, разделенных пробелами. В таблице радиостанций изменить частоту вещания радиостанции с заданным названием.