

Exercício - Agregação

Evandro César Freiburger

Instituto Federal de Educação, Ciência e Tecnologia

evandro.freiberger@cba.ifmt.edu.br

22 de junho de 2017

Exercício 01

1) Uma pesquisa sobre algumas características físicas da população de uma determinada região coletou os seguintes dados, referentes a cada habitante, para serem analisados: Sexo (masculino, feminino), Cor dos olhos (azuis, verdes, castanhos), Cor dos cabelos (louros, castanhos, pretos), Idade em anos. O programa deve perguntar no final de cada habitante, se o usuário deseja digitar os dados de um novo habitante.

Ao final da leitura dos dados, o programa deve exibir os seguintes dados:

- Maior e menor idade entre os habitantes pesquisados;
- Média de idade;
- Porcentagem de indivíduos do sexo feminino cuja idade está entre 18 e 35 anos inclusive e que tenham olhos verdes e cabelos louros;
- Número de habitantes que tenham idade entre 20 e 35 anos, inclusive e a porcentagem que esse número representa no total de habitantes pesquisados;

Exercício 01

Sugestão de resolução:

Definir uma classe Pessoa, contendo os atributos que armazene as informações: sexo, cor dos olhos, cor dos cabelos e idade.

Como o programa deve ler uma quantidade indefinida desse conjunto de dados, será necessário usar uma estrutura de dados linear para armazenar cada ocorrência de Pessoa, por exemplo, um vetor ou ArrayList, sendo o processamento feito por meio dessa lista.

O uso de enumerações é recomendado para representar as possibilidades de cor dos olhos, cor de cabelos e sexo.

Resolução 01 - Exercício 01

Criar um projeto denominado ProjetoPesquisa00. No projeto crie dois pacotes: aplicacao e negocio

Implemente as Enumerações (Enum em Java)

```
1 package negocio;  
2  
3 public enum CorCabelo {  
4  
5     CASTANHO,  
6     PRETO,  
7     LOIRO,  
8     RUIVO  
9 }
```

```
1 package negocio;  
2  
3 public enum CorOlho {  
4  
5     CASTANHO,  
6     AZUL,  
7     VERDE,  
8     PRETO  
9  
10 }
```

Resolução 01 - Exercício 01

Implemente a Enumeração (Enum em Java)

```
1 package negocio;  
2  
3  
4 public enum Sexo {  
5  
6     FEMININO,  
7     MASCULINO  
8  
9 }
```

Resolução 01 - Exercício 01 I

Implementação da Classe de Negócio Pessoa

```
1 package negocio;  
2 public class Pessoa {  
3  
4     private Sexo sexo;  
5     private CorOlho corOlhos;  
6     private CorCabelo corCabelo;  
7     private int idade;  
8  
9     public Pessoa(){  
10         this.sexo = Sexo.MASCULINO;  
11         this.corCabelo = CorCabelo.CASTANHO;  
12         this.corOlhos = CorOlho.CASTANHO;  
13     }  
14  
15     public Pessoa(Sexo sexo, CorCabelo corCabelo, CorOlho corOlho, int idade){  
16         this.setSexo(sexo);  
17         this.setCorOlhos(corOlho);  
18         this.setCorCabelo(corCabelo);  
19         this.setIdade(idade);  
20     }  
21  
22     public Sexo getSexo() {  
23         return sexo;  
24     }  
25  
26     public final void setSexo(Sexo sexo) {
```

Resolução 01 - Exercício 01 II

```
27     this.sexo = sexo;
28 }
29
30 public CorOlho getCorOlhos() {
31     return corOlhos;
32 }
33
34 public final void setCorOlhos(CorOlho corOlhos) {
35     this.corOlhos = corOlhos;
36 }
37
38 public CorCabelo getCorCabelo() {
39     return corCabelo;
40 }
41
42 public final void setCorCabelo(CorCabelo corCabelo) {
43     this.corCabelo = corCabelo;
44 }
45
46 public int getIdade() {
47     return idade;
48 }
49
50 public final void setIdade(int idade) {
51     this.idade = idade;
52 }
53 }
```

Resolução 01 - Exercício 01 I

Implementação da Classe de Negócio PesquisaPessoa

```
1 package negocio;  
2 public class PesquisaPessoa {  
3  
4     private Pessoa listaPessoa [];  
5     private int numPessoa;  
6  
7     public PesquisaPessoa() {  
8         this.numPessoa = 0;  
9         this.listaPessoa = new Pessoa[100];  
10    }  
11  
12    public void adicionarPessoa(Pessoa pessoa) {  
13        if (this.numPessoa < 100) {  
14            this.listaPessoa[this.numPessoa] = pessoa;  
15            this.numPessoa++;  
16        } else {  
17            System.out.println("Capacidade ultrapassada de armazenamento");  
18        }  
19    }  
20  
21    public Pessoa[] obterListaPessoa() {  
22        return this.listaPessoa;  
23    }  
24  
25    public int obterNumeroTotalPessoa() {  
26        return this.numPessoa;  
27    }  
28 }
```


Resolução 01 - Exercício 01 II

```
27     }
28
29     public int obterMaiorIdade() {
30         int maior = 0;
31         if (this.numPessoa > 0) {
32             maior = this.listaPessoa[0].getIdade();
33         }
34
35         for (int x = 0; x < this.numPessoa; x++) {
36             if (this.listaPessoa[x].getIdade() > maior) {
37                 maior = this.listaPessoa[x].getIdade();
38             }
39         }
40         return maior;
41     }
42
43     public int obterSomaldade() {
44         int soma = 0;
45         for (int x = 0; x < this.numPessoa; x++) {
46             soma += this.listaPessoa[x].getIdade();
47         }
48         return soma;
49     }
50
51     public float obterMedialdade() {
52         return this.obterSomaldade() / this.numPessoa;
53     }
54
55     public int obterPessoaPorFaixaIdade2035() {
```

Resolução 01 - Exercício 01 III

```
56     int resposta = 0;
57
58     for (int x = 0; x < this.numPessoa; x++) {
59         if (this.listaPessoa[x].getIdade() >= 20 && this.listaPessoa[x].
60             getIdade() <= 35) {
61             resposta++;
62         }
63     }
64     return resposta;
65 }
66
67 public int obterPessoaPorFaixaIdade1835_F_Verde_Louro() {
68     int resposta = 0;
69
70     for (int x = 0; x < this.numPessoa; x++) {
71         if (this.listaPessoa[x].getSexo() == Sexo.FEMININO) {
72             if (this.listaPessoa[x].getIdade() >= 18 && this.listaPessoa[x].
73                 getIdade() <= 35) {
74                 if (this.listaPessoa[x].getCorOlhos() == CorOlho.VERDE &&
75                     this.listaPessoa[x].getCorCabelo() == CorCabelo.LOIRO) {
76                     resposta++;
77                 }
78             }
79         }
80     }
81     return resposta;
82 }
```

Resolução 01 - Exercício 01 I

Implementação da Classe de Execução

```
1 package aplicacao;  
2  
3 import negocio.CorCabelo;  
4 import negocio.CorOlho;  
5 import negocio.PesquisaPessoa;  
6 import negocio.Pessoa;  
7 import negocio.Sexo;  
8  
9 public class Principal1 {  
10  
11     public static void main(String args[]) {  
12  
13         Pessoa pessoa;  
14         PesquisaPessoa pesquisa = new PesquisaPessoa();  
15         int totalPessoas, contador1;  
16  
17         pessoa = new Pessoa();  
18         pessoa.setSexo(Sexo.FEMININO);  
19         pessoa.setCorCabelo(CorCabelo.PRETO);  
20         pessoa.setIdade(10);  
21         pessoa.setCorOlhos(CorOlho.AZUL);  
22         pesquisa.adicionarPessoa(pessoa);  
23         //  
24         pessoa = new Pessoa();  
25         pessoa.setSexo(Sexo.FEMININO);  
26         pessoa.setCorCabelo(CorCabelo.LOIRO);
```

Resolução 01 - Exercício 01 II

```
27 pessoa.setIdade(30);
28 pessoa.setCorOlhos(CorOlho.VERDE);
29 pesquisa.adicionarPessoa(pessoa);
30 //
31 pessoa = new Pessoa();
32 pessoa.setSexo(Sexo.FEMININO);
33 pessoa.setCorCabelo(CorCabelo.LOIRO);
34 pessoa.setIdade(19);
35 pessoa.setCorOlhos(CorOlho.VERDE);
36 pesquisa.adicionarPessoa(pessoa);
37 //
38 pessoa = new Pessoa();
39 pessoa.setSexo(Sexo.MASCULINO);
40 pessoa.setCorCabelo(CorCabelo.LOIRO);
41 pessoa.setIdade(50);
42 pessoa.setCorOlhos(CorOlho.VERDE);
43 pesquisa.adicionarPessoa(pessoa);
44
45 System.out.println("Maior Idade: "+pesquisa.obterMaiorIdade());
46 System.out.println("Media de Idade: "+pesquisa.obterMedialdade());
47
48 totalPessoas = pesquisa.obterNumeroTotalPessoa();
49 contador1 = pesquisa.obterPessoaPorFaixaIdade2035();
50
51 System.out.println("Idade entre 20 e 35: "+contador1+" - "+(contador1 *
52     100 / totalPessoas)+"%");
53
54 contador1 = pesquisa.obterPessoaPorFaixaIdade1835_F_Verde_Louro();
```

Resolução 01 - Exercício 01 III

```
55     System.out.println("Feminino, Idade entre 18 e 35 Olhos Verdes e Cabelos  
56         Loiros: "+contador1+" - "+(contador1 * 100 / totalPessoas)+"%");  
57 }
```

Resolução 02 - Exercício 01 I

Criar um projeto denominado ProjetoPesquisa01. No projeto crie dois pacotes: aplicacao e negocio

Implementação da Classe de Negócio PesquisaPessoa

As demais classes do projeto são iguais à Resolução 01

```
1 package negocio;  
2  
3 import java.util.ArrayList;  
4  
5 public class PesquisaPessoa {  
6  
7     private ArrayList<Pessoa> listaPessoa;  
8  
9     public PesquisaPessoa() {  
10         this.listaPessoa = new ArrayList();  
11     }  
12  
13     public void adicionarPessoa(Pessoa pessoa) {  
14         this.listaPessoa.add(pessoa);  
15     }  
16  
17     public ArrayList<Pessoa> obterPessoas() {
```

Resolução 02 - Exercício 01 II

```
18     return this.listaPessoa;
19 }
20
21 public int obterNumeroTotalPessoa() {
22     return this.listaPessoa.size();
23 }
24
25 public int obterMaiorIdade() {
26     int maior = 0;
27     if (this.listaPessoa.size() > 0) {
28         maior = this.listaPessoa.get(0).getIdade();
29     }
30
31     for (Pessoa pessoaTemp : this.listaPessoa) {
32         if (pessoaTemp.getIdade() > maior) {
33             maior = pessoaTemp.getIdade();
34         }
35     }
36     return maior;
37 }
38
39 public int obterSomaldade() {
40     int soma = 0;
41     for (Pessoa pessoaTemp1 : this.listaPessoa) {
42         soma = soma + pessoaTemp1.getIdade();
43     }
44     return soma;
45 }
46 }
```

Resolução 02 - Exercício 01 III

```
47 public float obterMedialdade() {
48     return this.obterSomaldade() / this.listaPessoa.size();
49 }
50
51 public int obterPessoaPorFaixaIdade2035() {
52     int resposta = 0;
53
54     for (Pessoa pessoaTemp1 : this.listaPessoa) {
55         if (pessoaTemp1.getIdade() >= 20 && pessoaTemp1.getIdade() <= 35) {
56             resposta++;
57         }
58     }
59     return resposta;
60 }
61
62 public int obterPessoaPorFaixaIdade1835_F_Verde_Louro() {
63     int resposta = 0;
64
65     for (Pessoa pessoaTemp1 : this.listaPessoa) {
66         if (pessoaTemp1.getSexo() == Sexo.FEMININO) {
67             if (pessoaTemp1.getIdade() >= 18 && pessoaTemp1.getIdade() <= 35) {
68                 {
69                     if (pessoaTemp1.getOlhos() == CorOlho.VERDE && pessoaTemp1.
70                         getCabelo() == CorCabelo.LOIRO) {
71                         resposta++;
72                     }
73                 }
74             }
75         }
76     }
77 }
```


Resolução 02 - Exercício 01 IV

```
74     return resposta;  
75 }  
76 }
```

Resolução 03 - Exercício 01 I

Criar um projeto denominado ProjetoPesquisa02. No projeto crie dois pacotes: aplicacao e negocio

Implementação da Classe de Negócio PesquisaPessoa

```
1 package negocio;
2
3 import java.util.ArrayList;
4 public class PesquisaPessoa {
5     private ArrayList<Pessoa> lista;
6
7     public PesquisaPessoa() {
8         this.lista = new ArrayList();
9     }
10
11     public void adicionarPessoa(Pessoa pessoa) {
12         this.lista.add(pessoa);
13     }
14
15     public ArrayList<Pessoa> obterListaPessoa(){
16         return this.lista;
17     }
18
19     public int obterNumeroTotalPessoa(){
20         return this.lista.size();
21     }
22 }
```

Resolução 03 - Exercício 01 II

```
22
23 public int obterMaiorIdade( ArrayList<Pessoa> listaTemp) {
24     int maior = 0;
25     if( listaTemp.size() > 0){
26         maior = listaTemp.get(0).getIdade();
27     }
28     for ( Pessoa pessoaTemp : listaTemp) {
29         if ( pessoaTemp.getIdade() > maior) {
30             maior = pessoaTemp.getIdade();
31         }
32     }
33     return maior;
34 }
35
36 public int obterSomaldade( ArrayList<Pessoa> listaTemp) {
37     int soma = 0;
38     for ( Pessoa pessoaTemp1 : listaTemp) {
39         soma = soma + pessoaTemp1.getIdade();
40     }
41     return soma;
42 }
43
44 public float obterMedialdade( ArrayList<Pessoa> listaTemp) {
45     return this.obterSomaldade(listaTemp) / listaTemp.size();
46 }
47
48 public ArrayList<Pessoa> obterPessoaPorSexo( Sexo sexo , ArrayList<Pessoa>
49     listaTemp) {
50     ArrayList<Pessoa> resposta = new ArrayList();
```

Resolução 03 - Exercício 01 III

```
50     for (Pessoa pessoaTemp : listaTemp) {
51         if (pessoaTemp.getSexo() == sexo) {
52             resposta.add(pessoaTemp);
53         }
54     }
55     return resposta;
56 }
57
58 public int obterNumeroPessoaPorSexo(Sexo sexo, ArrayList<Pessoa> listaTemp) {
59     return this.obterPessoaPorSexo(sexo, listaTemp).size();
60 }
61
62 public ArrayList<Pessoa> obterPessoaPorCorOlho(CorOlho corOlho, ArrayList<
63     Pessoa> listaTemp) {
64     ArrayList<Pessoa> resposta = new ArrayList();
65     for (Pessoa pessoaTemp1 : listaTemp) {
66         if (pessoaTemp1.getCorOlhos() == corOlho) {
67             resposta.add(pessoaTemp1);
68         }
69     }
70     return resposta;
71 }
72
73 public int obterNumeroPessoaPorCorOlho(CorOlho corOlho, ArrayList<Pessoa>
74     listaTemp) {
75     return this.obterPessoaPorCorOlho(corOlho, listaTemp).size();
76 }
```

Resolução 03 - Exercício 01 IV

```
76 public ArrayList<Pessoa> obterPessoaPorCorCabelo(CorCabelo corCabelo ,
77     ArrayList<Pessoa> listaTemp) {
78     ArrayList<Pessoa> resposta = new ArrayList();
79     for (Pessoa pessoaTemp1 : listaTemp) {
80         if(pessoaTemp1.getCorCabelo() == corCabelo){
81             resposta.add(pessoaTemp1);
82         }
83     }
84     return resposta;
85 }
86
87 public int obterNumeroPessoaPorCorCabelo(CorCabelo corCabelo , ArrayList<
88     Pessoa> listaTemp) {
89     return this.obterPessoaPorCorCabelo(corCabelo , listaTemp).size();
90 }
91
92 public ArrayList<Pessoa> obterPessoaPorFaixaIdade(int idadeInicial , int
93     idadeFinal , ArrayList<Pessoa> listaTemp) {
94     ArrayList<Pessoa> resposta = new ArrayList();
95     for (Pessoa pessoaTemp1 : listaTemp) {
96         if(pessoaTemp1.getIdade() >= idadeInicial && pessoaTemp1.getIdade()
97             <= idadeFinal){
98             resposta.add(pessoaTemp1);
99         }
100     }
101     return resposta;
102 }
```

Resolução 03 - Exercício 01 V

```
00 public int obterNumeroPessoaPorFaixaIdade(int idadeInicial, int idadeFinal,
01     ArrayList<Pessoa> listaTemp) {
02     return this.obterPessoaPorFaixaIdade(idadeInicial, idadeFinal, listaTemp)
03         .size();
04 }
```

Implementação da Classe de Execução Principal1

```
1 package aplicacao;  
2  
3 import java.util.ArrayList;  
4 import negocio.CorCabelo;  
5 import negocio.CorOlho;  
6 import negocio.PesquisaPessoa;  
7 import negocio.Pessoa;  
8 import negocio.Sexo;  
9  
10 public class Principal1 {  
11  
12     public static void main(String args[]){  
13  
14         Pessoa pessoa;  
15         PesquisaPessoa pesquisa = new PesquisaPessoa();  
16         int totalPessoas, contador1;  
17         ArrayList<Pessoa> listaTemp;  
18  
19         pessoa = new Pessoa();  
20         pessoa.setSexo(Sexo.FEMININO);  
21         pessoa.setCorCabelo(CorCabelo.PRETO);  
22         pessoa.setIdade(10);  
23         pessoa.setCorOlhos(CorOlho.AZUL);  
24         pesquisa.adicionarPessoa(pessoa);  
25         //
```

Resolução 03 - Exercício 01 II

```
26  pessoa = new Pessoa();
27  pessoa.setSexo(Sexo.FEMININO);
28  pessoa.setCorCabelo(CorCabelo.LOIRO);
29  pessoa.setIdade(30);
30  pessoa.setCorOlhos(CorOlho.VERDE);
31  pesquisa.adicionarPessoa(pessoa);
32  //-----
33  pessoa = new Pessoa();
34  pessoa.setSexo(Sexo.FEMININO);
35  pessoa.setCorCabelo(CorCabelo.LOIRO);
36  pessoa.setIdade(19);
37  pessoa.setCorOlhos(CorOlho.VERDE);
38  pesquisa.adicionarPessoa(pessoa);
39  //-----
40  pessoa = new Pessoa();
41  pessoa.setSexo(Sexo.MASCULINO);
42  pessoa.setCorCabelo(CorCabelo.LOIRO);
43  pessoa.setIdade(50);
44  pessoa.setCorOlhos(CorOlho.VERDE);
45  pesquisa.adicionarPessoa(pessoa);
46
47  System.out.println("Maior Idade: "+pesquisa.obterMaiorIdade(pesquisa.
    obterListaPessoa()));
48  System.out.println("Média de Idade: "+pesquisa.obterMediaIdade(pesquisa.
    obterListaPessoa()));
49  totalPessoas = pesquisa.obterNumeroTotalPessoa();
50  contador1 = pesquisa.obterNumeroPessoaPorFaixaIdade(20, 35, pesquisa.
    obterListaPessoa());
```


Resolução 03 - Exercício 01 III

```
51 System.out.println("Idade entre 20 e 35: "+contador1+" - "+(contador1 *  
52 100 / totalPessoas)+"%");  
53  
54 listaTemp = pesquisa.obterPessoaPorFaixaIdade(18, 35, pesquisa.  
55 obterListaPessoa());  
56 listaTemp = pesquisa.obterPessoaPorSexo(Sexo.FEMININO, listaTemp);  
57 listaTemp = pesquisa.obterPessoaPorCorOlho(CorOlho.VERDE, listaTemp);  
58 contador1 = pesquisa.obterNumeroPessoaPorCorCabelo(CorCabelo.LOIRO,  
59 listaTemp);  
60 System.out.println("Feminino, Idade entre 18 e 35 Olhos Verdes e Cabelos  
61 Loiros: "+contador1+" - "+(contador1 * 100 / totalPessoas)+"%");  
62 }  
63 }
```

Resolução 03 - Exercício 01 I

Implementação da Classe de Execução Principal2

```
1 package aplicacao;  
2  
3 import java.util.ArrayList;  
4 import javax.swing.JOptionPane;  
5 import negocio.CorCabelo;  
6 import negocio.CorOlho;  
7 import negocio.PesquisaPessoa;  
8 import negocio.Pessoa;  
9 import negocio.Sexo;  
10  
11 public class Principal2 {  
12  
13     public static void main(String[] args) {  
14  
15         PesquisaPessoa pesquisa = new PesquisaPessoa();  
16         Pessoa pessoa;  
17  
18         int totalPessoas, contador1;  
19         ArrayList<Pessoa> listaTemp;  
20         int resp = 0;  
21         int idade;  
22         CorOlho corOlho;  
23         CorCabelo corCabelo;  
24         Sexo sexo;  
25     }
```

Resolução 03 - Exercício 01 II

```
26 do {
27     sexo = (Sexo) JOptionPane.showInputDialog(null, "Escolha o Sexo", "
28         Escolha",
29         JOptionPane.QUESTION_MESSAGE, null, Sexo.values(), Sexo.
30             values()[0]);
31     corOlho = (CorOlho) JOptionPane.showInputDialog(null, "Escolha a cor
32         dos olhos", "Escolha",
33         JOptionPane.QUESTION_MESSAGE, null, CorOlho.values(), CorOlho
34             .values()[0]);
35     corCabelo = (CorCabelo) JOptionPane.showInputDialog(null, "Escolha a
36         cor dos cabelos", "Escolha",
37         JOptionPane.QUESTION_MESSAGE, null, CorCabelo.values(),
38             CorCabelo.values()[0]);
39     idade = Integer.parseInt(JOptionPane.showInputDialog("Forneça a
40         Idade - em anos"));
41
42     pessoa = new Pessoa();
43     pessoa.setSexo(sexo);
44     pessoa.setCorOlhos(corOlho);
45     pessoa.setCorCabelo(corCabelo);
46     pessoa.setIdade(idade);
47
48     pesquisa.adicionarPessoa(pessoa);
49
50     resp = JOptionPane.showConfirmDialog(null, "Deseja fornecer dados de
51         outra pessoa", "Responda",
52         JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE);
53 } while (resp == JOptionPane.YES_OPTION);
```

Resolução 03 - Exercício 01 III

```
47 totalPessoas = pesquisa.obterNumeroTotalPessoa();
48 System.out.println("Total de Pessoas: "+totalPessoas);
49 System.out.println("Maior Idade: "+pesquisa.obterMaiorIdade(pesquisa.
    obterListaPessoa()));
50 System.out.println("Media de Idade: "+pesquisa.obterMedialdade(pesquisa.
    obterListaPessoa()));
51 contador1 = pesquisa.obterNumeroPessoaPorFaixaIdade(20, 35, pesquisa.
    obterListaPessoa());
52 System.out.println("Idade entre 20 e 35: "+contador1+" - "+(contador1 *
    100 / totalPessoas)+"%");
53
54 listaTemp = pesquisa.obterPessoaPorFaixaIdade(18, 35, pesquisa.
    obterListaPessoa());
55 listaTemp = pesquisa.obterPessoaPorSexo(Sexo.FEMININO, listaTemp);
56 listaTemp = pesquisa.obterPessoaPorCorOlho(CorOlho.VERDE, listaTemp);
57 contador1 = pesquisa.obterNumeroPessoaPorCorCabelo(CorCabelo.LOIRO,
    listaTemp);
58 System.out.println("Feminino, Idade entre 18 e 35 Olhos Verdes e Cabelos
    Loiros: "+contador1+
59     " - "+(contador1 * 100 / totalPessoas)+"%");
60 }
61 }
```

2) Um comerciante deseja desenvolver o levantamento do lucro das mercadorias que ele comercializa. Para isto, mandou digitar um conjunto de dados para cada uma das mercadorias que ele comercializa, são eles: nome do produto, preço de compra, preço de venda, quantidade comprada e quantidade vendida. Desenvolver um programa que:

- Determine e escreva o valor total de Compra;
- Valor total de venda;
- Lucro total;
- Mercadoria mais comprada;
- Mercadoria menos comprada;
- Mercadoria mais vendida;
- Mercadoria menos vendida.

Exercício 03 I

3) Para se determinar o número de lâmpadas necessárias para cada cômodo de uma residência, existem normas que dão o mínimo de potência de iluminação exigida por metro quadrado (m^2) conforme a utilização deste cômodo.

Seja a seguinte tabela tomada como exemplo:

UTILIZAÇÃO	POTÊNCIA / M^2
Quarto	15
sala de TV	15
Salas	18
Cozinha	18
varandas	18
escritórios	20
banheiro	20

O programa deve:

Exercício 03 II

- ❶ Ler a potencia da lâmpada que será usada em todos os cômodos. O número de lâmpadas de um cômodo será obtido pela divisão do total de potencia do cômodo pela potência da lâmpada adotada;
- ❷ A quantidade de cômodos que serão digitados;
- ❸ Leia um número indeterminado de linhas contendo cada uma:
 - Tipo de cômodo;
 - As dimensões do cômodo (largura e comprimento);
- ❹ Calcule e escreva para cada cômodo:
 - O tipo do cômodo;
 - A área do cômodo;
 - A potência de iluminação necessária;
 - O número de lâmpadas necessárias;
- ❺ Calcule e escreva para a residência:
 - O total de lâmpadas;
 - O total de potência;

Observações:

- 1 Se o número calculado de lâmpadas for fracionário, considerar o menor inteiro que contenha esse número. Ex $8,3 \Rightarrow 9$; $8,7 \Rightarrow 9$;
- 2 Modele a solução desse problema utilizando os conceitos de orientação a objetos que foram trabalhados em sala até o presente momento;

4) Uma universidade deseja fazer um levantamento a respeito de seu concurso de vestibular. Para cada curso, é fornecido o seguinte conjunto de valores:

- Código do curso;
- Número de vagas;
- Número de candidatos do sexo feminino;
- Número de candidatos do sexo masculino.

Desenvolver um programa que:

- Calcule e escreva, para cada curso, o número de candidatos por vaga e a porcentagem de candidatos do sexo feminino e masculino, (escreva também o código correspondente do curso);

Exercício 04 II

- Determine o maior número de candidatos por vaga e escreva esse número juntamente com o código do curso correspondente (supor que não haja empate)
- Calcule e escreva o total de candidatos.

5) O sistema de avaliação de uma determinada disciplina obedece aos seguintes critérios:

- Durante o semestre são aplicadas três provas;
- A nota final é obtida pela média aritmética das notas das provas;
- É considerado aprovado, o aluno que obtiver a nota final superior ou igual a 6 e que tiver comparecido a um mínimo de 75% das aulas.

Desenvolver um programa que:

- Leia o total de aulas ministradas para a turma.
- Leia um conjunto de dados contendo um número de matrícula, as três notas e a frequência (número de aulas frequentadas) para cada aluno.
- Calcule:
 - Nota final do aluno;
 - Maior e menor nota da turma;

Exercício 05 II

- Nota média da turma;
 - Total de alunos reprovados por nota;
 - Total de alunos reprovados por falta;
 - Percentual de alunos reprovados por nota.
 - Percentual de alunos reprovados por faltas.
- Escreva:
 - Para cada aluno, o número de matrícula, a frequência, a nota e o resultado (aprovado ou reprovado);
 - Maior e menor nota da turma;
 - Nota média da turma;
 - Total de alunos reprovados por nota;
 - Total de alunos reprovados por faltas;
 - Percentual de alunos reprovados por nota;
 - Percentual de alunos reprovados por faltas;