

## Primeiro Trabalho de Tópicos em Programação I

Prof. Flávio Miguel Varejão

### I. Descrição do Problema

A empresa *Seu Imóvel S.A.* é uma corretora especializada na venda de terrenos e residências do tipo casa e do tipo apartamento. Ela oferece aos clientes uma ampla gama de imóveis, os quais se encontram registrados no seu catálogo de produtos.

Todas as entradas desse catálogo possuem como características comuns um identificador único e o nome do proprietário.

Além dessas características comuns, terrenos possuem a identificação do seu tipo predominante de solo (arenoso, argiloso, rochoso), o preço do metro quadrado do terreno e podem ter formato triangular, retangular e trapezoidal. Nos terrenos triangulares são registrados no catálogo os tamanhos de sua base e altura, em metros. Nos terrenos retangulares são registrados os tamanhos dos dois lados diferentes, em metros. Nos terrenos trapezoidais são registrados os tamanhos de suas duas bases e de sua altura, também em metros.

Já residências possuem como características comuns o número de quartos, o número de vagas de garagem e o preço do metro quadrado de área construída. Casas possuem ainda identificação do número de pavimentos, sua área construída por pavimento e sua área livre, ambas em metros quadrados, assim como o preço do metro quadrado de área livre. Apartamentos possuem identificação do andar onde está localizado, sua área construída, uma indicação se possui área de lazer e o número total de andares do prédio.

O preço de cada imóvel é calculado segundo fórmulas que variam de acordo com seu tipo. No caso de terrenos, o preço é calculado pela fórmula:

$$\text{preço} = \text{preçoMetroQuadradoTerreno} * \text{área} * \text{fatorMultiplicativo}$$

O *fatorMultiplicativo* varia de acordo com o tipo predominante de solo. Se o solo é arenoso, o fator multiplicativo é 0,9; se o solo é argiloso, é 1,3; e se o solo é rochoso, é 1,1.

No caso de casas, o preço é calculado pela fórmula:

$$\text{preço} = \text{preçoMetroQuadradoAreaConstruida} * \text{areaPavimento} * \text{numeroPavimentos} + \text{preçoMetroQuadradoAreaLivre} * \text{areaLivre}$$

No caso de apartamentos, o preço é calculado pela fórmula:

$$\text{preço} = \text{preçoMetroQuadradoAreaConstruida} * \text{areaConstruida} * (0,9 + \text{andarApto} / \text{totalAndaresPrédio}) * \text{fatorLazer}$$

O *fatorLazer* vale 1,15 se o prédio conta com área de lazer e 1, em caso contrário.

O proprietário da *Seu Imóvel S.A.* resolveu implantar um sistema de informação para controlar melhor o catálogo da empresa. Para tanto, ele decidiu te contratar para

implementar uma versão preliminar desse sistema. O sistema deve atender a seguinte especificação:

## II. Especificação do Sistema

### Funcionalidades a serem implementadas:

1. Leitura de dados do catálogo existente de um arquivo texto, denominado "catalogo.txt".
2. Leitura de operações de atualização, sobre o arquivo catálogo, de um arquivo texto, denominado "atual.txt".
3. Leitura de um arquivo texto, denominado "espec.txt", especificando dados relativos aos resultados a serem apresentados.
4. Obtenção dos seguintes resultados:
  - a) Lista dos imóveis mais caros em ordem crescente de preço. A quantidade de imóveis que devem fazer parte da lista será definida através da especificação de um percentual no arquivo "espec.txt".
  - b) Lista dos terrenos argilosos de menor área em ordem decrescente de área. A quantidade de imóveis que devem fazer parte da lista será definida através da especificação de um percentual no arquivo "espec.txt".
  - c) Lista contendo as casas com área construída superior a um limite especificado no arquivo "espec.txt" e com preço inferior a um limite também especificado no arquivo "espec.txt". A lista deve ser ordenada decrescentemente por número de quartos.
5. Gravação do valor do resultado da soma dos identificadores do  $i$ -ésimo imóvel da lista do item  $a$ , do  $j$ -ésimo imóvel da lista do item  $b$  e do  $k$ -ésimo imóvel da lista do item  $c$  e dos identificadores numéricos dos imóveis das listas dos itens  $a$ ,  $b$  e  $c$  na saída padrão. Os valores de  $i$ ,  $j$  e  $k$  são especificados no arquivo "espec.txt"

### Observações:

(i) Na possibilidade remota de haver empate entre os elementos dos itens  $a$ ,  $b$  e  $c$ , deve-se utilizar o identificador numérico único como critério de desempate. Assim, no caso de empate em uma ordenação crescente, será colocado à frente na lista o elemento de menor identificador numérico. Em caso de ordenação decrescente, ocorre o inverso, isto é, fica à frente o elemento de maior identificador numérico.

(ii) Deve-se considerar que os elementos das listas sempre começam a ser contados a partir da posição 1 e não da posição 0.

(iii) Em caso de não existir o elemento desejado na lista (por exemplo, se o valor  $i$ ,  $j$  ou  $k$  for igual a zero ou superior ao índice do último elemento da lista), considerar o valor zero na soma para o item correspondente.

(iv) Cálculo do número de imóveis correspondente ao percentual lido:

$$\text{numeroImoveis} = \text{parteInteira}(\text{percentualLido} * \text{totalImoveis} / 100)$$

(v) Cálculo do número de terrenos argilosos correspondente ao percentual lido:

$$\text{numeroArgilosos} = \text{parteInteira}(\text{percentualLido} * \text{totalArgilosos} / 100)$$

(vi) Considere que o catálogo da empresa possui no máximo 300 imóveis.

#### Formato dos Dados do Sistema:

Identificador: inteiro não negativo  
Nome Proprietário: até 40 caracteres  
Tipo de Solo: 1 caractere (A, G ou R)  
Dimensões Terrenos: ponto flutuante de precisão simples  
Número Quartos: inteiro não negativo  
Número Vagas: inteiro não negativo  
Número Pavimentos: inteiro não negativo  
Área Pavimento: ponto flutuante de precisão simples  
Área Livre: ponto flutuante de precisão simples  
Andar: inteiro não negativo  
Área Constr: ponto flutuante de precisão simples  
Lazer: 1 caractere (S ou N)  
Número Andares: inteiro não negativo  
Preço M<sup>2</sup> Terreno: inteiro não negativo  
Preço M<sup>2</sup> Construída: inteiro não negativo  
Preço M<sup>2</sup> Livre: inteiro não negativo  
Preço: ponto flutuante de precisão simples

#### Entrada de Dados:

A entrada de dados será realizada a partir de três arquivos texto (***catalogo.txt*** com dados do catálogo de produtos; ***atual.txt***, com dados sobre as operações a serem realizadas sobre o catálogo; e ***espec.txt***, com dados a respeito do relatório a ser gerado pelo programa). Por simplificação, cada registro em cada um destes arquivos será separado por uma linha em branco. Além disso, cada dado de um registro destes arquivos corresponderá a uma linha do arquivo e nunca faltará qualquer dado.

A seguir, apresentam-se as especificações dos registros de cada um destes arquivos com pequenos exemplos correspondentes.

##### ***catalogo.txt***

##### Especificação do registro:

<categoria> (triang, retang, trapez, casa, apto)  
<identificador>  
<proprietario>  
<<dados\_categoria>> % dados especificos da categoria

Campos específicos da categoria terreno triangular:

<tipo de solo>  
<preco metro quadrado terreno>  
<base>  
<altura>

Campos específicos da categoria terreno retangular:

<tipo de solo>

<preco metro quadrado terreno>  
<lado1>  
<lado2>

Campos específicos da categoria terreno trapezoidal:

<tipo de solo>  
<preco metro quadrado terreno>  
<base1>  
<base2>  
<altura>

Campos específicos da categoria casa:

<numero de quartos>  
<numero de vagas>  
<numero de pavimentos>  
<area pavimento>  
<preco metro quadrado area pavimento>  
<area livre>  
<preco metro quadrado area livre>

Campos específicos da categoria apartamento:

<numero de quartos>  
<numero de vagas>  
<andar>  
<area construida>  
<preco metro quadrado area construida>  
<lazer>  
<numero andares>

Exemplo de arquivo:

triang  
1763  
Fernando Gonzalez  
G  
123  
30.3  
10.5

retang  
2323  
José da Silva  
R  
254  
20.3  
30.1

casa

3294  
Maria da Penha  
4  
2  
2  
150  
2000  
300  
276

trapez  
1722  
Mauricio Rocha  
G  
128  
15.4  
24.2  
17.4

apto  
3217  
Silvana Rossi  
4  
3  
7  
200  
4300  
S  
12

***atual.txt***

Especificação do registro:

<operação> % i(inclusão);e(exclusão);a(alteração)  
<dados da operação> % dados específicos para operação

Nas operações de inclusão e alteração, os dados da categoria têm formato idêntico aos dos registros do arquivo catalogo. No caso da alteração, o identificador único designará o membro a ser alterado. Todos os dados do produto serão alterados (com exceção do identificador, por motivos óbvios). Na operação de exclusão, o único dado adicional é o identificador do produto que será excluído.

Exemplo de arquivo:

i

trapez  
1777  
Mauro Galvao  
A  
180  
13.4  
20.2  
12.4

e  
3217

a  
casa  
3294  
Maria da Penha  
4  
2  
3  
150  
1500  
300  
700

***espec.txt***

Especificação do registro:

<percentual de imóveis caros>	% inteiro não negativo
<percentual de menores argilosos >	% inteiro não negativo
<area limite>	% ponto flutuante de precisão simples
<preco limite>	% ponto flutuante de precisão simples
<valor de i>	% inteiro não negativo
<valor de j>	% inteiro não negativo
<valor de k>	% inteiro não negativo

Exemplo de arquivo:

40  
70  
180.00  
100000.00  
3  
1  
4

\*\*\* Atenção: Para execução no boca, as trilhas para a localização dos arquivos serão lidas da entrada padrão, uma por linha, na sequência *catalago.txt*, *atual.txt* e *espec.txt*. Por exemplo, uma possível entrada dessas trilhas seria:

```
/home/boca/catalogo.txt
/home/boca/atual.txt
/home/boca/espec.txt
```

#### Saída de Dados:

A apresentação dos resultados será feita pela saída padrão. A primeira linha conterá o valor inteiro positivo da soma especificada no item 5 das descrições das funcionalidades do sistema, na segunda linha os identificadores da lista do item *a* separados por vírgula, na terceira linha os identificadores da lista do item *b* separados por vírgula e na quarta linha os identificadores da lista do item *c* separados por vírgula.

#### Especificação do registro:

<valor da soma>	% inteiro não negativo
<lista de identificadores>	% inteiros não negativos
<lista de identificadores>	% inteiros não negativos
<lista de identificadores>	% inteiros não negativos

#### Exemplo de saída:

```
1352
1378, 2198, 988
2121, 77, 654, 536, 736
29, 1
```

### **III. Requisitos da implementação**

1. Modularize seu código adequadamente. O uso de variáveis globais é proibido, mas constantes globais são permitidas;
2. Crie códigos claros e organizados. Utilize um estilo de programação consistente, Comente seu código.
3. Os arquivos *catalogo.txt*, *atual.txt* e *espec.txt* devem ser lidos na mesma pasta onde se encontra os arquivos fonte e executáveis do seu programa. O arquivo *saida.txt* deve ser gerado nesta mesma pasta.
4. Atente para o fato de que não é suficiente fazer um programa que produza as respostas esperadas. A avaliação do programa também levará em conta a modularização utilizada, o reaproveitamento de código e o uso apropriado de memória.

### **IV. Condições de Entrega**

O trabalho deve ser feito individualmente e submetido por e-mail até as 23:59 horas da data limite especificada (28/10/2019). Note que a data limite já leva em conta um dia adicional de tolerância para o caso de problemas de submissão via rede. Isso significa que o aluno deve submeter seu trabalho até no máximo um dia antes da data limite. Se o aluno resolver submeter o trabalho na data limite, estará fazendo isso assumindo o risco do trabalho ser cadastrado no sistema após o prazo. Em caso de recebimento do trabalho após a data limite, o trabalho não será avaliado e a nota será ZERO. Aluno que receber zero por este motivo e vier pedir para o professor considerar o trabalho estará cometendo um ato de DESRESPEITO ao professor e estará sujeito a perda adicional de pontos na média.

## **V. Formato de Entrega dos Trabalhos**

O recebimento dos trabalhos é automatizado. Portanto, as regras a seguir devem ser seguidas à risca para evitar que seu trabalho não possa ser avaliado.

O código-fonte de sua solução deverá ser compactado e entregue por e-mail (anexo ao e-mail) para o endereço [fvarejao@gmail.com](mailto:fvarejao@gmail.com).

Serão aceitos trabalhos entregues até as 23h59 da data limite (28/10/2019). O assunto do e-mail deverá ser o seguinte:

`tp1:trab<1>:<nome>:`

O termo “<id>” deve ser substituído pelo número correspondente do trabalho (1). O termo “<nome>” deverá ser substituído pelo nome e o último sobrenome do aluno, sem acentos, til ou cedilha, como no exemplo abaixo:

`tp1:trab1:Flavio Varejao:`

O arquivo compactado deve estar no formato tar.gz com o nome trab1.tar.gz e conter apenas os arquivos fonte do programa (não deve conter executáveis ou arquivos compilados). Para isso, abra um console, mude o diretório de trabalho para a pasta onde se encontra o código-fonte do trabalho e execute o seguinte comando (no caso do trabalho 1):

`tar -zcvf trab1.tar.gz *`

Preste bastante atenção para fazer com que o código fonte não seja colocado em subdiretórios dentro do arquivo compactado.

Um exemplo de um e-mail de envio do trabalho:

Para: [fvarejao@ninf.inf.ufes.br](mailto:fvarejao@ninf.inf.ufes.br)  
De: Joao da Silva  
Assunto: tp1:trab1:Joao Silva:  
Anexo: trab1.tar.gz

Para a compilação e execução dos programas, em princípio, será utilizada a versão instalada no labgrad do gcc. Caso haja alguma modificação de versão de correção, ela será divulgada oportunamente. Os programas serão compilados e executados no sistema operacional linux. Para que não haja problemas na correção do seu trabalho e você seja prejudicado, garanta que ele é compilado na versão de correção e executado no sistema operacional linux.



## **VI. Avaliação**

Os trabalhos terão nota zero se:

- A data de entrega for fora do prazo estabelecido;
- O trabalho não compilar;
- O trabalho não gerar o arquivo com o resultado e formato esperado;
- For detectada a ocorrência de plágio pelo sistema.

Ainda, os trabalhos poderão ser avaliados segundo os seguintes critérios:

- Cumprimento das restrições estabelecidas no ítem III deste documento;
- Modularização (considerando o uso de arquivos separados para os diversos tipos abstratos de dados);
- Ausência de uso de variáveis globais;
- Legibilidade (nomes de variáveis bem escolhidos, código bem formatado, uso de comentários quando necessário, etc.);
- Consistência (utilização de um mesmo padrão de código);
- Eficiência (sem exageros, tentar evitar grandes desperdícios de recursos);

### **Observação importante**

**Caso haja algum erro neste documento, serão publicadas novas versões e divulgadas erratas em sala de aula. É responsabilidade do aluno manter-se informado, freqüentando as aulas ou acompanhando as novidades na página da disciplina na Internet.**