

Programação de Computadores II - BCC702

Lista de Exercícios

Classes

Exercício 01

Defina uma classe *Ponto*, que possui os atributos reais *x* e *y*, que representam as suas coordenadas no plano cartesiano.

Implemente as seguintes funções como membros da classe:

1. um construtor sem parâmetros, que inicializa os valores de *x* e *y* com 0;
2. um construtor que recebe como parâmetros dois valores reais a serem utilizados para inicializar *x* e *y*;
3. os métodos *get* e *set* para cada um dos atributos;
4. um método *imprimir*, que imprime as coordenadas do ponto no formato (*x*, *y*);
5. um método *distância*, que recebe como parâmetro um ponto e calcula a distância euclidiana do objeto até este ponto.

Na função *main*, crie dois pontos *p1* e *p2*, um utilizando cada um dos construtores. Utilize o método *set* para alterar as coordenadas de *p1*. Use o método *imprimir* para imprimir os dois pontos.

Exercício 02

Defina uma classe denominada *Elevador* para armazenar as informações de um elevador dentro de um prédio.

A classe deve armazenar o andar atual (0=térreo), total de andares no prédio, excluindo o térreo, capacidade do elevador, e quantas pessoas estão presentes nele.

O **método construtor** da classe deve receber como parâmetros a capacidade do elevador e o total de andares no prédio. Os elevadores sempre começam no térreo e vazios.

A classe deve também disponibilizar os seguintes métodos:

- **entra:** para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- **sai:** para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- **sobe:** para subir um andar (não deve subir se já estiver no último andar);
- **desce:** para descer um andar (não deve descer se já estiver no térreo);
- **get....:** métodos para obter cada um dos dados armazenados;
- **exibe:** para exibir o estado do elevador na tela.

O programa deve simular o funcionamento de um elevador que é operado por meio de um menu de opções:

1. Entra uma pessoa
2. Sai uma pessoa
3. Sobe um andar
4. Desce um andar
5. Termina

Em cada iteração o menu é apresentado ao usuário, que escolhe uma opção a ser executada, até que se escolha terminar. A cada execução deve ser apresentado o estado atualizado do elevador.

Exemplo de execução do programa:

```
SIMULAÇÃO DO ELEVADOR
-----
Número de andares: 5
Capacidade do elevador: 3
-----
[5 andares, 3 pessoas] andar 0 com 0 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 1
-----
[5 andares, 3 pessoas] andar 0 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 1
-----
[5 andares, 3 pessoas] andar 0 com 2 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
-----
[5 andares, 3 pessoas] andar 1 com 2 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 2
-----
[5 andares, 3 pessoas] andar 1 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
-----
[5 andares, 3 pessoas] andar 2 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
-----
[5 andares, 3 pessoas] andar 3 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 1
-----
[5 andares, 3 pessoas] andar 3 com 2 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 1
-----
[5 andares, 3 pessoas] andar 3 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 1
lotado!
-----
[5 andares, 3 pessoas] andar 3 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
-----
[5 andares, 3 pessoas] andar 4 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
-----
[5 andares, 3 pessoas] andar 5 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 3
último andar!
-----
[5 andares, 3 pessoas] andar 5 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 4
-----
[5 andares, 3 pessoas] andar 4 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 4
-----
[5 andares, 3 pessoas] andar 3 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina  => 4
-----
```

[5 andares, 3 pessoas] andar 2 com 3 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 2

[5 andares, 3 pessoas] andar 2 com 2 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 2

[5 andares, 3 pessoas] andar 2 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 4

[5 andares, 3 pessoas] andar 1 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 4

[5 andares, 3 pessoas] andar 0 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 4
térreo!

[5 andares, 3 pessoas] andar 0 com 1 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 2

[5 andares, 3 pessoas] andar 0 com 0 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 2
vazio!

[5 andares, 3 pessoas] andar 0 com 0 pessoas
Escolha uma opção: 1.entra 2.sai 3.sobe 4.desce 5.termina => 5
vazio!