

Atividade 1

Problemas propostos

Usando o NetBeans e criando um projeto Java Ant para cada situação, codifique programas Java que solucionem os problemas a seguir. Leia e releia com atenção os enunciados. Os exemplos são mais diretos que o desejável em implementação – ou seja, sinta-se à vontade para incluir mensagens de orientação ao usuário durante a execução do programa.

1. Relativo ao atendimento de eventos, o hotel necessita de uma funcionalidade que indique qual de seus dois auditórios é o mais adequado para um evento. O auditório Alfa conta com 150 lugares e espaço para até 70 cadeiras adicionais. O auditório Beta conta com 350 lugares, sem espaço para mais cadeiras. Desenvolva um programa Java que receba o número de convidados do evento e faça uma verificação sobre esse número: se for maior que 350 ou menor que zero, deverá ser mostrada a mensagem “Número de convidados inválido”. Se o valor informado é válido, deverá ser mostrado na tela qual dos auditórios é o mais adequado. No caso do auditório Alfa, será preciso calcular ainda quantas cadeiras adicionais serão necessárias, observando o limite citado anteriormente.

Exemplo:

```
[Entrada de dados]
190 convidados (número de convidados)
[Saída de dados]
Use o auditório Alfa
Inclua mais 40 cadeiras
```

2. Escreva um programa Java que faça a troca de quartos entre dois hospedes. É preciso que ele receba o nome de um cliente e sua idade; o cliente a princípio será hospedado no quarto A. Depois, o programa deve receber o nome de outro cliente e sua idade; se o cliente for mais jovem, ficará no quarto B, mas, se a idade deste for maior que a do primeiro cliente, ele ficará no quarto A e o primeiro cliente ficará no quarto B. Além disso, se a pessoa mais velha for idosa (com 60 anos ou mais),

terá desconto de 40%. O programa deve mostrar ao final o nome do cliente que ficou no quarto A e o desconto, se houver, e o nome do cliente que ficou no quarto B.

Exemplo:

```
[Entrada de dados]
Maria (nome cliente)
20 (idade cliente)
Joao (nome cliente)
60 (idade cliente)
[Saída de dados]
Quarto A: Joao com desconto de 40%; Quarto B:
Maria
```

3. Escreva um programa Java que receba o valor padrão de uma diária e em seguida receba vários nomes de hóspedes e suas idades. Caso a idade do hóspede seja menor de 4 anos, ele não paga hospedagem – nesses casos, é preciso que seja mostrada na tela a mensagem “(Nome do hóspede) possui gratuidade”. Hóspedes com mais de 80 anos pagam metade – então dever ser mostrado na tela “(Nome do hóspede) paga meia”. O usuário informará hóspedes até digitar a palavra “PARE”, que interrompe a entrada de dados. Ao fim, o programa deve mostrar a quantidade de gratuidades, a quantidade de meias hospedagens e o valor total, considerando todos os hóspedes informados.

Exemplo:

```
[Entrada de dados]
100 (R$ valor padrão da diária)
[Entrada de dados]
Adolfo (nome do hóspede)
50 (idade)
Murilo (nome do hóspede)
1 (idade)
[Saída de dados]
```

```
Murilo possui gratuidade
[Entrada de dados]
Joaquim (nome do hóspede)
85 (idade)
[Saída de dados]
Joaquim paga meia
[Entrada de dados]
PARE (nome do hóspede)
[Saída de dados]
Total de hospedagens: R$150; 1 gratuidade(s); 1
meia(s)
```

4. Monte um algoritmo em que o usuário poderá cadastrar e pesquisar hóspedes. O algoritmo deve oferecer um menu com três opções ao usuário: 1- cadastrar; 2- pesquisar; 3- sair. A opção “cadastrar” deve permitir que o usuário informe um nome de hóspede, gravando-o em memória (máximo de 15 cadastros; caso atinja essa quantidade, mostre “Máximo de cadastros atingido”). A opção “pesquisar” deve permitir que o usuário informe um nome e, caso seja encontrado um nome exatamente igual, mostre a mensagem “Hóspede (nome) foi encontrado no índice (índice onde foi cadastrado)”. Se o nome não foi encontrado, mostre “Hóspede não encontrado”. O algoritmo deve permitir que o usuário realize essas operações repetidas vezes, até que use a opção “3”, que encerra o algoritmo.

Exemplo:

```
[Saída de dados]
Digite 1- cadastrar; 2- pesquisar; 3- sair
[Entrada de dados]
1 (opção)
Maria da Silva (nome do hóspede)
[Saída de dados]
Digite 1- cadastrar; 2- pesquisar; 3- sair
[Entrada de dados]
1 (opção)
```

```
José Freitas (nome do hóspede)
[Saída de dados]
Digite 1- cadastrar; 2- pesquisar; 3- sair
[Entrada de dados]
2 (opção)
José Freitas
[Saída de dados]
Hóspede José Freitas foi encontrado no índice 1
[Saída de dados]
Digite 1- cadastrar; 2- pesquisar; 3- sair
[Entrada de dados]
3
```

5. Monte um programa Java que mostre na tela em formato de tabela os quartos ocupados e os desocupados. Considere que o hotel tem 4 andares e 3 quartos por andar. Primeiro, o usuário registrará os quartos ocupados, informando para cada ocupação o número do apartamento (andar de 1 a 4 e número de quarto de 1 a 3). O programa deverá questionar “Deseja informar outra ocupação? (S/N)” e o usuário poderá informar quantos quartos quiser, até que responda “N” a essa pergunta. Em seguida, o programa mostrará uma tabela, em que o primeiro andar é o inferior e o último, o superior, marcando com X o quarto ocupado.

Exemplo:

```
[Saída de dados]
Informe Andar e Quarto
[Entrada de dados]
1 (andar)
1 (quarto)
[Saída de dados]
Deseja informar outra ocupação? (S/N)
[Entrada de dados]
S
```

[Saída de dados]

Informe Andar e Quarto

[Entrada de dados]

2 (andar)

3 (quarto)

[Saída de dados]

Deseja informar outra ocupação? (S/N)

S

[Saída de dados]

Informe Andar e Quarto

[Entrada de dados]

4(andar)

1(quarto)

[Saída de dados]

Deseja informar outra ocupação? (S/N)

[Entrada de dados]

N

[Saída de dados]

Ocupação do hotel:

1º andar:

- quarto 1 ocupado
- quarto 2 desocupado
- quarto 3 desocupado

2º andar:

- quarto 1 desocupado
- quarto 2 desocupado
- quarto 3 ocupado

3º andar:

- quarto 1 desocupado
- quarto 2 desocupado
- quarto 3 desocupado

4º andar:

- quarto 1 ocupado

- quarto 2 desocupado
- quarto 3 desocupado