

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, sintase à vontade para incluir mensagens de orientação ao usuário durante a execução do programa.

1. Com relação ao atendimento a 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 a quantidade: se for maior que 350 ou se for menor que zero, mostre a mensagem “Número de convidados inválido”. Se o valor informado for válido, mostre na tela qual dos auditórios é o mais adequado. No caso do auditório Alfa, calcule ainda quantas cadeiras adicionais serão necessárias, observando o limite citado.

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 hóspedes. O programa deve receber o nome de um cliente e a idade dele (este cliente, a princípio, será hospedado no quarto A), depois receber o nome de outro cliente e a idade (este, se for mais jovem, ficará no quarto B, mas, se for mais velho que o primeiro cliente, ficará no quarto A e o primeiro cliente ficará no quarto B). Além disso, se a pessoa mais velha

for idosa (tem 60 anos ou mais), ela terá desconto de 40%. O programa deve mostrar, ao final, qual cliente ficou no quarto A e o desconto, se houver, e qual 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 o hóspede tenha idade menor que 4 anos, ele não paga hospedagem (nesse caso, o programa deve mostrar na tela “(Nome do hóspede) possui gratuidade”); mas, se o hóspede tiver mais de 80 anos, ele paga metade (o programa deve mostrar 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, o programa deve mostrar “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, o programa deve mostrar “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 quatro andares e três quartos por andar. Primeiro, o usuário registrará os quartos ocupados, informando para cada ocupação o número do apartamento (andares de 1 a 4 e números 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 é 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:

4º andar [X][][]

3º andar [][][]

2º andar [][][X]

1º andar [X][][]