Obs: Para todos os exercícios, criar os atributos privados e acessar os mesmos através de métodos públicos “getters and setters”. Quando necessário.

1. – Estruturas If-Else

Crie um novo Java Project que solicite três notas: de 0 a 10,0.

Considerando que media > 7 seja aprovado, desenvolva uma lógica que exiba os seguintes resutados:

Se media >= 7, "Parabéns: você foi aprovado!";

Else "Sinto muito: você foi reprovado."

2. – Estruturas If-Else

Crie um Java Project que solicite duas notas: de 0 a 10,0.

Exiba a média e o conceito correspondente, considerando que o mínimo para aprovação é 7,0

Considere também:

Se média < 7 "Insuficiente"

Se média < 8 “Suficiente"

Se média < 9.5 "Bom"

Se for maior que 9.5 “Excelente"

3. – Estruturas switch

* Crie um novo Java Project que solicite um número correspondente a de um mês, por exemplo (de 1 a 12).
* Grave a entrada do usuário variável do tipo byte para o Valor do MÊS.
* Crie uma lógica para exibir o nome do mês correspondente ao número informado pelo usuário.

Se o número for menor que 1 ou maior que 12, exiba uma mensagem de erro.

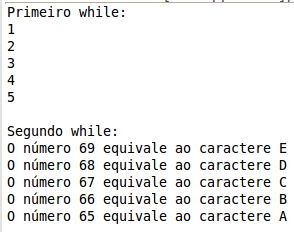
4. – While

Crie um novo Java Project, chamado EstruturaWhileUm. Na classe principal, crie um laço que imprima o valor de uma variável inteira

* Esta variável deve iniciar com zero.
* A variável deve ser incrementada antes de ser impressa.
* Isso deve se repetir enquanto a variável for menor que 5

Depois, escreva um laço que imprima o valor de uma variável inteira e o caractere correspondente a ele.

* Esta variável deve iniciar com 69.
* A variável deve ser decrementada depois de ser impressa.
* Isso deve se repetir enquanto a variável for maior ou igual a 65.

O resultado pode ser disponibilizado da seguinte forma:

5. – Switch

Crie um novo Java Project que solicite o nome de um usuário. **Quando** o nome for válido, exiba uma mensagem inidicando o cenário Ok.

Quando o nome for inválido, exiba uma mensagem de erro.

Obs:

Para ser válido, o nome deve ter de 5 a 50 caracteres

Dica: Enquanto o nome\_valido for FALSE, continua executando o programa.

6. – Do

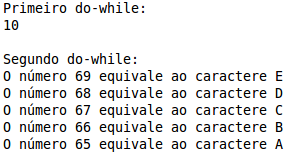
Crie um novo Java Project atribuindo um nome qualquer, e uma classe Principal. Escreva um laço que imprima o valor de uma variável inteira.

* Esta variável deve iniciar com nove.
* A variável deve ser incrementada antes de ser impressa.
* Isso deve se repetir enquanto a variável for menor que cinco.

Após isso, escreva um laço que imprima o valor de uma variável inteira e o caractere correspondente a ele.

* Esta variável deve iniciar com 69.
* A variável deve ser decrementada depois de ser impressa.
* Isso deve se repetir enquanto a variável for maior ou igual a 65.

Exemplo da saída:



7. – do-while

Crie um novo Java Project e atribua um nome qualquer. Crie uma classe Principal que solicite o e-mail de um usuário.

**Quando** o e-mail for válido, exiba uma mensagem que indique sua validade.

**Quando** o e-mail for inválido, solicite-o novamente.

Observações:

Para ser válido, o e-mail deve atender aos seguintes requisitos:

* Deve ter pelo menos 5 caracteres.
* Deve ter somente um símbolo de arroba.
* Este símbolo não pode ser o primeiro caractere.
* Este símbolo não pode ser o último caractere.

8. – For

Crie um novo Java Project no Eclipse, atribua o nome ExercicioForUm. Crie uma classe Principal e escreva um laço que imprima o valor de uma variável inteira.

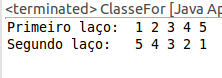
Observações:

* Esta variável deve iniciar com 1.
* A variável deve ser incrementada depois de ser impressa.
* Esse processo deverá repetir **enquanto** a variável for menor ou igual a 5.

Escreva outro laço que imprima o valor de uma variável inteira.

* Esta variável deve iniciar com 5.
* A variável deve ser decrementada depois de ser impressa.
* Isso deve se repetir **enquanto** a variável for maior ou igual a 1.

O resultado deverá ser igual ao seguinte:

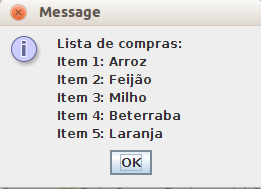


9. – For

Crie um novo Java Project no Eclipse, atribua o nome ExercicioForDois. Crie uma classe Principal e solicite a descrição de 5 itens para compor uma lista de compras.

* Grave todos os itens em uma única string.
* Ao final, exiba a lista de compras com os 5 itens registrados.

Observação. Não utilizar nenhum tipo de List ou ArrayList. Apenas utilizar a concatenação de String.



10. – Break

Crie um novo Java Project no eclipse com o nome ExercicioBreak. Após isso, criar uma classe Principal que solicite o nome do usuário.

* Quando o nome for válido, exiba uma mensagem com o nome informado
* Quando o nome for inválido, solicite-o novamente.
* Para ser válido, o nome deve ter pelo menos 2 caracteres.
* Para repetir a solicitação do nome, utilize um laço cuja condição jamais se torne falsa.

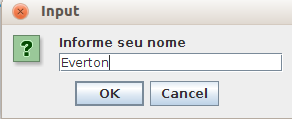
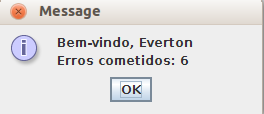
**Interrompa** este laço quando o nome for válido.

11. – Continue

Crie um novo Java Project e atribua o nome ExercicioContinue. Após isso, crie um programa que solicite o nome do usuário.

* Utilize um laço for para realizar esta solicitação.
* Quando o nome for inválido, solicite-o novamente.
* Utilize o contador do laço for para contar quantas vezes um nome inválido foi informado.
* Se nenhum erro tiver sido cometido, apenas apresente uma mensagem de boas-vindas.
* Se um nome inválido tiver sido informado, a mensagem de boasvindas

deve incluir o número de erros cometidos.

12. – ForEach - Vetor

Crie um novo Java Project chamado ForEach. Crie uma classe Principal que possua um vetor capaz de armazenar nomes de até 10 amigos.

* Utilize um laço for para captar e armazenar os nomes.
* Utilize um laço for sem contador para recuperar e exibir a lista de amigos (laço foreach).

13. – ForEach - ArrayList

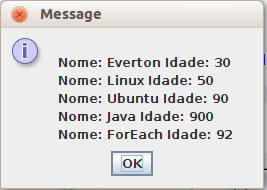
Crie um novo Java Project chamado ForEachDois. Crie os seguintes pacotes:

* br.com.treinamento.hp.beans
* br.com.treinamento.hp.main

No pacote beans, crie a classe Pessoa com os atributos privados String nome e Integer idade.

No pacote main, crie a classe Principal com um ArrayList<> de Pessoas.

Utilizando o FOR comum, implemente uma regra para cadastrar 5 pessoas e adicione na lista

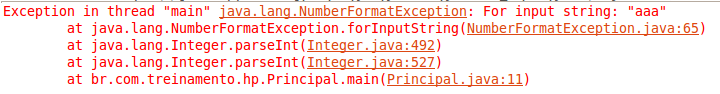


14. – Exceções

Crie um novo Java Project e atribua um nome Qualquer. Crie uma classe Principal que deve solicitar um número inteiro e exibir o seu cubo.

* Utilize um diálogo gráfico para solicitar e exibir os dados. (JOptionPane)
* Execute o aplicativo e informe um dado que **não possa ser convertido** para o tipo int.

Resultado:



15. – Exceções – Try Catch

Crie um Novo Java Project e atrbua um nome qualquer. Crie a classe Principal de modo que a mesma receba dois números e faça a divisão do primeiro pelo segundo.

* Utilizar diálogos gráficos para captar os números e exibir o resultado. (JOptionPane)
* Se ocorrer uma exceção durante a execução destas instruções, o programa deve exibir uma mensagem de erro.

A mensagem de erro deve conter:

* O nome da classe que representa a exceção.
* A mensagem de erro padrão desta exceção.