

LAB01-A: Objetos e Classes

INSTRUÇÕES INICIAIS:

1. Obtenha o arquivo lab01-a.zip (disponível no ensino aberto);
2. Descompacte esse arquivo no seu home-dir;
3. Siga as instruções (instrucoes.pdf Slides-Eclipse.pdf) para a criação de um projeto Java no IDE Eclipse (disponível no EA).

Nota: Os exercícios devem ser executados na ordem apresentada, pois o nível de dificuldade é crescente.

CONHECENDO UM POUCO MAIS SOBRE O AMBIENTE ECLIPSE E A LINGUAGEM JAVA:

O **Eclipse** é um ambiente (IDE – Integrated Development Environment) para desenvolvimento de programas, inicialmente desenvolvida pela IBM e hoje, disponibilizado como uma ferramenta *Open-Source* pelo consórcio chamado “Eclipse.org”. O **Workspace** é responsável por administrar os recursos do usuário que são organizados em um ou mais projetos. Em **File** → **Switch workspace** você poderá alterar a área de trabalho do seu **Eclipse**. Os projetos em Java normalmente são organizados em pacotes (**packages**). Os pacotes são estruturas de diretórios utilizadas para organizar classes e interfaces. Eles fornecem um mecanismo para reutilização de software que ajuda a prevenir o conflito de nomes. A Sun Microsystems, criadora da linguagem Java, especificou uma convenção para nomes de pacotes. Cada nome de pacote deve iniciar com seu nome de domínio Internet, na ordem inversa, seguido do nome do pacote. Portanto, no nosso curso, iremos usar sempre: br.unicamp.ic.mc302.<nome do pacote>

CONHECENDO UM POUCO SOBRE Vetores em Java:

Vetores são coleções de objetos ou tipos primitivos. Os tipos devem ser conversíveis ao tipo em que foi declarado o vetor:

```
int[] vetor = new int[10];
```

Cada elemento do vetor é inicializado a um valor *default*, dependendo do tipo de dados (null, para objetos; 0, para int, long, short, byte, float, double; Unicode 0, para char e **false**, para boolean). Elementos podem ser recuperados a partir da posição 0:

```
int elemento_1 = vetor[0];  
int elemento_2 = vetor[1];
```

Vetores podem ser inicializados no momento em que são criados. Exemplo:

```
String[] semana = {"Dom", "Seg", "Ter", "Qua", "Qui", "Sex", "Sab"};
```

Todo vetor em Java possui a propriedade **length** que informa o número de elementos que possui, extremamente útil em blocos de repetição:

```
for (int x = 0; x < vetor.length; x++) {  
    vetor[x] = x*x;  
}
```

Uma vez criados, vetores não podem ser redimensionados.

Java possui uma coleção de APIs (bibliotecas) padrão, organizadas em pacotes, que podem ser usadas para construir aplicações. O **ArrayList** faz parte do JavaCollections Framework (JCF). Esta classe é bastante útil pois permite criar vetores dinâmicos e ainda oferece métodos úteis para manipular o conteúdo do mesmo. Para se utilizar esta classe basta colocar no início do seu arquivo:

```
import java.util.ArrayList;
```

Pacote br.unicamp.ic.mc302.hello:

1. Abra o arquivo Hello.java. Esse arquivo define uma classe chamada **Hello** que implementa apenas um método, **main()**. O método **main()** define um programa principal, ou seja, um programa que pode ser chamado a partir do sistema operacional (similarmente à função **main()** da linguagem C). A assinatura do método deve ser idêntica à que é apresentada no arquivo **Hello.java**. Compile esse arquivo. Instruções sobre como compilar um arquivo estão disponíveis no ensino aberto.

Pacote *br.unicamp.ic.mc302.documento*:

2. Abra o arquivo `Documento.java`, estude o código e, em seguida, compile-o.
3. Abra o arquivo `ExemploDocumento.java`. Esse arquivo define um programa principal que cria um objeto do tipo `Documento` e envia mensagens para ele. Compile a classe `ExemploDocumento` e execute-a.
4. Modifique a classe `ExemploDocumento` para que, após enviar a mensagem `imprimir()`, envie também a mensagem `editar()`. Compile e execute o programa novamente.
5. Altere a operação `imprimir()` da classe `Documento` para que, ao invés de imprimir uma mensagem padrão, imprima os valores dos atributos `autor` e `dataChegada` do objeto. Altere a classe `ExemploDocumento` para criar um segundo objeto do tipo `Documento` (`d2`) e imprimir as informações dos objetos `d1` e `d2`. Compile e execute a versão modificada de `ExemploDocumento`.

Pacote *br.unicamp.ic.mc302.contaCor*:

6. Abra os arquivos `ContaCor.java` e `Exemplo.java`, compile-os e execute a classe `Exemplo`.
7. Modifique a visibilidade do atributo `saldoAtual` da classe `ContaCor` para `public`. Em seguida, altere a operação `main()` da classe `Exemplo` para que, além de creditar R\$100,00 através da operação `creditaValor()`, credite mais R\$100,00 adicionando diretamente ao valor do atributo `saldoAtual`. Qual é a sua conclusão?
8. Na classe `ContaCor`, crie uma operação `transferir()` para realizar transferências entre contas. Essa operação deve receber como argumentos um outro objeto do tipo `ContaCor` que corresponde ao beneficiado, um `double` representando o valor a ser transferido, e a senha da conta que terá o saldo reduzido. A operação deve devolver `void` como resultado. A implementação da operação `transferir()` deve usar os métodos já definidos na classe `ContaCor`.
9. Altere o código da operação `main()` da classe `Exemplo` para que crie um outro objeto do tipo `ContaCor`, referenciado por uma variável chamada `c2`. Insira na operação `main()`, código para realizar uma transferência de R\$100,00 de `c1` para `c2`, usando a operação `transferir()` de `c1`. Inclua também código para imprimir o saldo de `c2` ao final da operação. Compile e execute a classe `Exemplo`.
10. Implemente em Java uma classe chamada `MeuVetor` com operações para inicializar cada elemento do vetor e para intercalar 2 vetores. Crie um programa principal que instancie 2 vetores com 4 elementos e depois os intercala. Você pode definir outras operações se necessário.