

MAC0321

Laboratório de Programação Orientada a Objetos

Denis Mauá
IME-USP

Baseado fortemente no material didático do Prof. Fabio Kon



Aula 4

Construtores e especificadores de acesso

Pacotes em Java

- Agrupar classes relacionadas
 - Similar a ideia de pasta de arquivos
- Evita conflito de nomes e permite limitar acesso
- Biblioteca de classes de Java (built-in)
 - `java.lang.Math`
 - `java.util.Vector`
- Pacotes definidos por usuários
 - `conversao.ConversorDeTemperatura`
 - `conversao.ConversorDeDistancia`
 - `br.usp.ime.BibliotecaDeMusicas`

Pacotes em Java

- Para usar classes em um pacote, precisamos importá-las
 - `import` pacote.subpacote.Classe
 - `import` pacote.* // todas as classes

```
import java.util.Scanner;

class MyClass {
    public static void main(String[] args) {
        Scanner myObj = new Scanner(System.in);
        System.out.println("Enter username");

        String userName = myObj.nextLine();
        System.out.println("Username is: " + userName);
    }
}
```

Pacotes em Java

- Para definir um novo pacote, usamos a palavra `package` na primeira linha do arquivo-fonte:
 - `package` nomedopacote

```
// Arquivo usp/poligonos/Polígono.java
package usp.polígonos;

class Polígono {
    int[] _lados;
    ...
}
```

```
// Arquivo usp/sim/App.java
import usp.polígonos.*;

class App {
    public static void main(String[] args) {
        Polígono p = new Polígono(3, 4);
        ...
    }
}
```

Encapsulamento

- Organização de dados e métodos que atuam sobre eles em uma única unidade independente
- **Objetivo:** facilitar modularidade e separar especificação e implementação (abstração)
 - Permite alterar implementação de forma segura
 - Permite **esconder informação** e **restringir acesso** (nome de atributos não precisam ser divulgados a usuários do código)

Especificadores de Acesso

- **Public** – qualquer objeto
- **Private** – apenas por objetos da própria classe
- **Protected** – objetos no mesmo pacote ou objetos especializados (no mesmo ou em outro pacote)
- **Default** – objetos no mesmo pacote

métodos de acesso getters e setters

- Não é uma boa ideia acessar diretamente os atributos das classes
 - Viola encapsulamento
- Solução comum em Java:
 - todos atributos são private (ou protected)
 - criação de getters e setters
 - Exemplo, se o atributo é int x
 - int getX()
 - void setX(int valor)

final

- `final float pi = 3.1415926538;`
- `final String siglaUniversidade = "USP";`
- `final int maxAlunos = 40;`
- qualquer tentativa de mudar o valor dessas variáveis dará erro de compilação.

static

- **static antes de um atributo**
 - indica que aquela é uma variável da classe e não do objeto
 - todas as instâncias compartilham a mesma variável
 - espaço na memória é alocado quando a classe é carregada no sistema
- **static antes de um método**
 - indica que aquele é um método da classe
 - ele pode ser chamado diretamente através da classe, sem a necessidade de se criar uma instância.
 - Exemplo: **Math.sin(2*pi);**

Resumo

- Pacotes (package)
- especificadores de acesso
- métodos de acesso
- static
- final