INSTITUTO FEDERAL
Santa Catarina

PROGRAMAÇÃO ORIENTADA A OBJETOS

Samuel da Silva Feitosa

INSTITUTO FEDERAL
Santa Catarina

REVISÃO

- Modificadores de acesso:
  - **public**, **private**, protected
- Analogias com a vida real
  - Telefone público, celular, telefone fixo
- É muito comum, e faz todo sentido, que seus **atributos** sejam **private** e quase todos seus **métodos** sejam **public** (não é uma regra!).

INSTITUTO FEDERAL
Santa Catarina

ENCAPSULAMENTO

- Uma das **ideias mais importantes** da orientação a objetos é o encapsulamento.
- Significa **separar o programa em partes**, o mais isoladas possível.
- Encapsular também significa **esconder a implementação** dos objetos.

"O princípio de esconder a estrutura de dados utilizada e somente prover uma interface bem definida é chamado de encapsulamento."

INSTITUTO FEDERAL
Santa Catarina

BENEFÍCIOS

- **Modularidade:** o código-fonte para um objeto pode ser escrito e mantido **independentemente** do código-fonte de outros objetos (desacoplamento). Além disso, como não dependem de outros objetos, cada objeto pode ser utilizado livremente no sistema.
- **Ocultação de informações:** um objeto possui uma **interface pública** que outros objetos podem utilizar para comunicarem-se com ele. Mas o objeto pode manter informações **privadas** e métodos podem ser modificados em qualquer momento sem afetar os outros objetos que dependem dele.

INSTITUTO FEDERAL
Santa Catarina

EXEMPLOS

INSTITUTO FEDERAL
Santa Catarina

SOBRECARGA DE MÉTODOS

INSTITUTO FEDERAL
Santa Catarina

SOBRECARGA

- Quando **dois ou mais métodos** são definidos na **mesma classe** com o **mesmo nome**, dizemos que houve uma **sobrecarga de métodos**.
- Uma sobrecarga de métodos **só é válida** se as **listas de parâmetros** dos métodos **são diferentes** entre si.
- Exemplo: aumento de salário de uma pessoa.

INSTITUTO FEDERAL
Santa Catarina

CONSTRUTORES

INSTITUTO FEDERAL
Santa Catarina

CONSTRUTORES

- Um **construtor** permite que um determinado trecho de código seja executado **toda vez que um objeto é criado**, ou seja, toda vez que o operador `new` é chamado.
- Assim como os métodos, os construtores **podem receber** parâmetros.
- Diferentemente dos métodos, os construtores **não devolvem** resposta.

INSTITUTO FEDERAL
Santa Catarina

GETTERS E SETTERS

INSTITUTO FEDERAL
Santa Catarina

GET e SET

- O modificador **private** faz com que ninguém consiga modificar, nem mesmo ler, o atributo em questão.
- Precisamos então **arranjar uma maneira** de fazer esse acesso.
- Sempre que precisamos arrumar uma maneira de fazer alguma coisa com um objeto, **utilizamos de métodos**.

INSTITUTO FEDERAL
Santa Catarina

GET e SET

- Na linguagem Java, há uma **convenção de nomenclatura** para os métodos que têm como finalidade **acessar ou alterar** as propriedades de um objeto.
- Os nomes dos métodos que permitem a **consulta das propriedades** de um objeto devem possuir o prefixo **get**.
- Os nomes dos métodos que permitem a **alteração das propriedades** de um objeto devem possuir o prefixo **set**.

INSTITUTO FEDERAL
Santa Catarina

EXERCÍCIO

INSTITUTO FEDERAL
Santa Catarina

EXERCÍCIO

Implementar as classes ***Data*** e ***Estudante*** conforme diagramas UML:

- A classe ***Data*** deve permitir somente a entrada de datas válidas, ou seja, o método *setData()* **deve validar** se os números de entrada correspondem a dias, meses e anos válidos. **Não esqueça** de validar a quantidade de dias de cada mês, considerando se é ou não um ano bisexto.
- A classe ***Estudante*** deve seguir a especificação. Notem que existem dois métodos *calculaMedia()* (sobrecarga), onde o primeiro deve apenas calcular a **média aritmética** entre as três notas, e o segundo permite adicionar uma **pontuação extra** sobre a média.
- Implemente no método *main()* da classe ***Principal*** várias chamadas para os diversos métodos criados.