

## ESQUEMA DA AULA PRÁTICA 7

### □ Herança

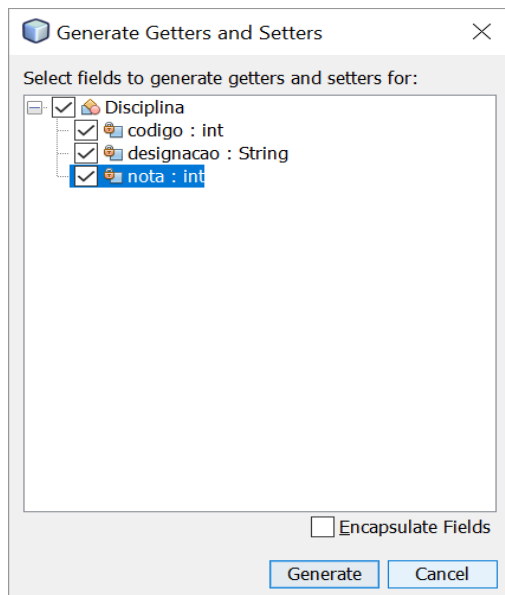
1 - Implemente uma classe `Disciplina` com os atributos `codigo` (valor do tipo `int`) `designacao` (valor do tipo `String`) e `nota` (`int`) e com os seguintes métodos:

a) Construtor com os atributos `codigo` e `designacao` como parâmetros;

*Agora que já sabe como fazer os getters e setters pode começar a usar algumas funcionalidades do seu editor que lhe permitem desenvolver código com mais rapidez:*

b) Para construir os getters e setters para os atributos da classe `Disciplina` faça:

Dentro da classe `Disciplina`, click no botão direito do rato, seleccione “Insert code” / “Getter and Setter...” / seleccione todos os atributos e click no botão `Generate` (ver figura).



- Verifique que os getters e setters da classe foram gerados de forma automática.

b) Repita o processo, agora para o método toString. Depois de gerado o código poderá formatar a String como quiser.

c) Construa você mesmo o método equals.

d) Construa o método clone .

e) Construa o Construtor de cópia

**2** - Implemente ainda uma classe Aluno tal que cada objecto do tipo Aluno tenha um **número e um nome** de aluno. Defina um construtor com o número e nome de aluno como parâmetros assim como os getters e setters para cada atributo.

**3** – A partir da classe Aluno pretendem definir-se duas subclasses, AlunoLicenciatura e AlunoPosGraduacao.

A classe AlunoLicenciatura terá como atributos o **curso** frequentado (String) e uma **lista** de Disciplinas com objectos do tipo Disciplina. Esta lista (objecto do tipo ArrayList<Disciplina>) irá conter as disciplinas feitas pelo aluno.

- Defina a classe AlunoLicenciatura com um construtor que tem como parâmetros um objecto do tipo Aluno e o curso e com os seguintes métodos:

a) Getters e Setters. Use o processo automático apenas para os tipos primitivos. Para os tipos referenciados (arrays e objectos) deve construir os métodos forma a não copiar apenas o endereço dos objectos.

b) toString

c) equals

d) clone

e) addDisciplina que dado um objecto do tipo Disciplina deverá adicioná-lo à lista de *disciplinas* caso este não exista ainda na lista.

f) getNotaDisciplina que dado um código de disciplina deve devolver a nota da disciplina. Se essa disciplina não constar da lista de disciplinas do aluno de licenciatura o método deve devolver o valor 0.

g) método que calcule a média das classificações obtidas pelo aluno de licenciatura.

**4 -** Numa classe de teste:

a) Construa um método público e estático que dado um array de objectos do tipo Aluno conte quantos desses alunos são alunos de licenciatura.

b) Teste as operações:

- getNotaDisciplina da classe AlunoLicenciatura ;
- o método que calcula a média das classificações do aluno de licenciatura;
- o método que, dado um array de objetos do tipo aluno, conta quantos desses alunos são alunos de licenciatura.

c) Indique qual o output do programa anterior.

**5 -** Desenvolva um sistema de Gestão de Contas Bancárias<sup>1</sup>.

O mesmo deverá ser capaz de realizar diversas operações bancárias sobre contas, tendo em atenção que existem 3 tipos de conta diferentes: normal, vencimento e a prazo.

Todas as contas têm um número, um ou mais titulares, uma data de criação e um saldo.

As contas normais não permitem que o saldo seja inferior a zero.

As contas vencimento permitem que o saldo seja negativo (até determinado limite que pode ser alterado) e mantêm o número de identificação bancária (NIB) da entidade empregadora.

A conta a prazo rende juros, dependendo do tempo que cada depósito permaneça na conta.

Sobre todas as contas pretende-se ter disponíveis as seguintes operações:

- a. Criação de conta
- b. Levantamento de uma quantia
- c. Depósito de uma quantia
- d. Consulta do saldo atual
- e. Consulta da data, montante e descrição dos últimos 5 movimentos (depósitos e levantamentos)

Identifique os objetos relevantes para o seu sistema e implemente as classes necessárias servindo-se do mecanismo de herança sempre que apropriado.

**6 -** Teste as classes anteriores.

---

<sup>1</sup> Exercício adaptado de "Programação Orientada aos Objectos em Java 2", F. Mário Martins, FCA, 2000.