

ATIVIDADE

Assunto:

Polimorfismo.

Orientações:

A atividade deve ser executada individualmente e entregue através do ambiente *Google Classroom*.

Regras de criação dos programas:

Crie um novo projeto Java denominado **AtividadePolimorfismo**. As classes devem possuir os nomes informados no texto. Ao final, o projeto deve ser exportado para um arquivo em formato ZIP.

Nome completo:

Jorge Fernando Ramos Bezerra

1. Explique o que é polimorfismo e em quais situações é útil.
R: Se trata de uma característica da orientação a objeto na qual se admite tratamento idêntico para objetos diferentes baseado em relações de semelhança. É bastante útil quando queremos manipular um objeto independente de sua implementação, pois sabemos que tal característica já foi implementada na classe mais genérica.
2. O polimorfismo possui desvantagens? Explique.
R: A desvantagem é que acabamos não tratando o objeto como seu tipo específico o que acaba não sendo possível acessar métodos e atributos que não foram chamados na classe que ele herda.
3. O que é amarração tardia? Explique.
R: É a chamada de métodos em tempo de execução e não em tempo de compilação como acontece em alguns compiladores.
4. Para demonstrar o uso do polimorfismo, siga os passos a seguir:
 - a. Crie um projeto no Eclipse denominado AtividadePolimorfismo.
 - b. Crie a classe Produto com os atributos privados: nome, descrição e preço. Esta classe deve possuir os métodos get e set para cada atributo, bem como um construtor que receba todos os argumentos. Crie também os métodos equals e toString. O método toString deve imprimir uma frase parecida com a seguir: "Produto: <nome>, <descrição>, <preço>". Não deve existir construtor padrão.
 - c. Crie a classe Livro, que herda de Produto, adicionando os atributos privados: autores, ISBN e editora. Esta classe deve possuir os métodos get e set para cada atributo da classe Livro, bem como um construtor que receba todos os argumentos (incluindo os da classe Produto, que devem ser repassados para a classe pai no construtor). Os métodos equals e toString devem ser sobrescritos. O método toString deve imprimir uma frase parecida com a seguir: "Livro: <nome>, <descrição>, <preço>, <autores>, <ISBN>, <editora>".
 - d. Crie a classe Principal, cujo objetivo é realizar a interação com o usuário. Esta classe deve possuir um vetor (ou ArrayList) da classe Produto. Você deve criar um menu em modo texto em que o usuário pode selecionar entre as opções:
 - i. Cadastrar produto
 - ii. Listar produtos (deve utilizar o toString para imprimir)
 - iii. Cadastrar livro
 - iv. Listar livros (deve utilizar o toString para imprimir)
 - v. Imprimir tudo (deve utilizar o toString para imprimir)

vi. Sair

- e. Note que apesar do vetor/lista ser do tipo Produto, você conseguirá adicionar objetos do tipo Livro. Adicionalmente, quando chamar o método toString() dos objetos, algumas vezes será chamado o método da classe Produto e outras vezes o método da classe Livro, quando o objeto armazenado tiver sido um livro. Você economizou código? A estrutura do seu código ficou mais simples? Isto é possível graças ao polimorfismo. Quais suas conclusões sobre o polimorfismo?

R: Tem suas vantagens e desvantagens, vai depender muito do uso e da necessidade.

Boa sorte! Prof. Igor.