

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:

Mac Myller da Silva Carlos

1. Explique o que é polimorfismo e em quais situações é útil.
Resposta: Polimorfismo é o princípio a partir do qual as classes derivadas de uma única classe base são capazes de invocar os métodos que, embora apresentem a mesma assinatura, comportam-se de maneira diferente para cada uma das classes derivadas.
Ele é útil para facilitar a generalização de algoritmos e estruturas de dados.
2. O polimorfismo possui desvantagens? Explique.
Sim. O polimorfismo só é possível com a amarração tardia e ela pode deixar a execução do programa mais lento.
3. O que é amarração tardia? Explique.
Resposta: Em linguagens estruturadas, os compiladores realizam a amarração em tempo de compilação.
Em linguagens orientadas a objeto com polimorfismo, não temos como saber o tipo real do objeto em tempo de compilação
A amarração é feita em tempo de execução, isso que é a amarração tardia
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?

Boa sorte! Prof. Igor.