

LISTA DE EXERCÍCIO – SEMANA 07

Notas: - Pode ser utilizado como ajuda os livros contidos na bibliografia da disciplina.

- 1) Use o conceito de polimorfismo para implementar o programa abaixo:
 - Crie uma classe abstrata **Imovel** que possui um atributo do tipo **endereço** (rua, número, bairro, cidade, estado) e um atributo **preço**. Nessa classe você deverá também declarar o método abstrato *calcularValorImovel()*.
 - Crie duas classes filhas (subclasses) da classe **Imovel**:
 - a) **ImovelNovo**
 - b) **ImovelVelho**
 - Cada uma dessas classes deve conter um construtor que receba o endereço e preço do imóvel. Cada classe também deve implementar o método *calcularValorImovel()*, retornando ao usuário o valor final do imóvel (o cálculo de cada imóvel deverá ser elaborado por você).
 - No método main da classe Principal, crie um único *array* de tamanho 200 do tipo **Imovel**. Insira de forma alternada, instâncias dos dois tipos de imóveis neste *array*. Em seguida, use um laço de repetição para exibir o valor de cada imóvel, chamando o método *calcularValorImovel()* dos objetos. Utilize os métodos da classe Scanner para interação com o usuário.
 - Descreva as categorias de polimorfismo que você usou no exercício, indicando o nome da categoria (sobrecarga, coerção, inclusão e/ou paramétrico), a classe em que foi usada e a(s) instrução(ões) referentes a cada categoria.
- 2) O setor responsável pelos editais de monitoria estabeleceu que todos os monitores obrigatoriamente disponibilizar seus horários de atendimento aos alunos e tirar as dúvidas dos alunos utilizando o método que julgarem mais apropriado (ex.: mostrar exemplos, fornecer dicas, resolver exercícios). Além dos monitores selecionados, os integrantes do PET também oferecem esse tipo de atendimento e devem seguir as mesmas exigências. Em ambos os casos, cada estudante deve informar o nome, matrícula e disciplina na qual prestará atendimento. Foi definido ainda que somente os monitores selecionados pelo edital precisam enviar um relatório de atividades realizadas.
 - a. Faça a modelagem do programa utilizando o diagrama de classes da UML.
 - b. Após a modelagem, faça a implementação utilizando a linguagem Java.
 - c. Na classe que possui o método main(), implemente uma solução que utilize o polimorfismo de inclusão e coerção *downcast*.

- 3) Você está desenvolvendo um jogo de aventura com vários tipos de personagens. Cada personagem possui atributos básicos e um método para calcular o dano de ataque, mas esse cálculo varia dependendo do tipo de personagem.
- Crie uma classe abstrata **Personagem** com os seguintes atributos: nome, nível, força base e um método abstrato *calcularDano()*.
 - Crie três classes filhas (subclasses) da classe **Personagem**:
 - c) **Guerreiro**: o dano é calculado como $\text{força base} * \text{nível} * 1.2$
 - d) **Mago**: o dano é calculado como $\text{força base} * \text{nível} * 1.5 - 10$
 - e) **Arqueiro**: o dano é calculado como $\text{força base} * \text{nível} + 15$
 - Cada uma dessas classes deve ter um construtor completo que inicialize os atributos.
 - No método main da classe Principal, crie uma *Collection* do tipo *TreeSet* para armazenar elementos do tipo **Personagem**. Insira, de forma alternada, instâncias dos três tipos de personagem nessa coleção. Em seguida, use um laço de repetição para exibir o nome e o dano de cada personagem, usando o método *calcularDano()*. Os elementos devem ser exibidos em ordem decrescente de dano. Utilize os métodos da classe Scanner para interação com o usuário.