

Para os exercícios abaixo, crie um projeto e cada exercício deverá ficar em um pacote diferente.

Após finalizado o exercício, salve o projeto em um zip e envie-o pelo Moodle . Não será aceito o recebimento via e-mail. Plágio não será tolerado e tanto a pessoa que forneceu quanto a que utilizou perderão os pontos da prática.

Entrega até o final da aula: exercício 1

Entrega do restante: até 17/05

Exercício 1: Implemente as classes Conta, ContaCorrente e ContaPoupanca. A conta e ContaCorrente está descrita na aula sobre herança. A única diferença é que a classe Conta deverá ter um método setSaldo (para ser usado pela ContaPoupanca). Implemente a classe ContaPoupanca ela deverá ter o valor do rendimento de 0 até 1 representando o rendimento de 0 a 100%. O rendimento, juntamente com o número da conta, será inicializado no construtor e deverá ter seus métodos de alteração. Além disso, crie o método atualizarRendimento que atualiza o saldo a partir do rendimento que a poupança possui. Por exemplo, se o saldo é de R\$ 200,00 e o rendimento = 0.10 (10%), então após executar atualizarRendimento o saldo será de R\$ 220,00.

Crie uma classe teste que cria um vetor de 1000 contas poupanças $\text{deposite}(i+1)*100$ reais sendo que i é a posição desta conta no vetor. Logo após, atualize o rendimento destes objetos e, logo após, exiba os dados de todas essas contas.

Exercício 2: Em um Sistema de Gestão Hospitalar, considere as classes Médico e Paciente. As classes possuem características em comum, tais como: CPF, RG, nome, cidade, estado, telefone, estado civil. As seguintes operações são comuns a ambas classes: alteração e recuperação de dados. Entretanto, Médico e Paciente têm características diferentes. Por exemplo, Médico possui CRM e Especialidade, e paciente está associado a um convênio.

Implemente as classes Especialidade e Convênio (apenas com os dados código e descrição). Utilizando o recurso de herança, implemente as classes Pessoa, Médico e Paciente. Implemente o toString de todas as classes. O toString das classes Médico e Paciente deverão chamar o toString da classe Pessoa para obter a String dos dados em comum com a pessoa. Você deverá implementar a classe cidade, estado e telefone. Porém, para facilitar, use a implementação dessas classes de práticas anteriores.

Para testar, crie uma classe principal dois médicos, uma pessoa e dois pacientes. Cada médico terá uma especialidade distinta e todos os pacientes possuirão o mesmo convênio. Adicione-os em uma única lista de pessoas (`List<Pessoa>`). Note que você pode adicioná-los em uma mesma lista, pois todos são de uma superclasse comum e a lista é do tipo desta superclasse. Percorra essa lista e exiba seus dados.

--