

Programação Orientada a Objetos

EXERCÍCIO 23

Ler os capítulos 4 e 5 da apostila e fazer os respectivos exercícios.

EXERCÍCIO 24

Para evitar erros de digitação de sequências de números tais como: número de conta bancária, número de CPF, CNPJ e afins, geralmente adiciona-se ao número dígito(s) verificador(es). Por exemplo, no número de cpf "741.392.463-85", os números 8 e 5 são dígitos verificadores.

Modifique a classe `Funcionario` da apostila acrescentando um atributo `String cpf` e o método `boolean validaCPF()`. O método deverá validar CPF's nos formatos "12345678910" ou "123.456.789-10".

Pesquise:

1. <http://www.clubedainformatica.com.br/site/2003/10/22/algoritmo-de-cpf/>
2. O método `charAt` da classe `String` e o método `isDigit` da classe `Character`

EXERCÍCIO 25

Modifique o método `adiciona` da classe `Empresa` para adicionar ao vetor `empregados` apenas o objeto `funcionario` que tenha CPF válido.

EXERCÍCIO 26

Faça uma classe `Cartao` que tenha internamente um objeto do tipo `Conta`, uma senha numérica e a `relacaoDeDebitos`.

- a) `relacaoDeDebitos` é um vetor que armazena o nome do estabelecimento onde foi efetuado o débito e o valor do débito. Proponha uma classe para armazenar estas informações.
- b) Tenha um método `debitar`, que tem como parâmetros: o valor a ser debitado, o nome do estabelecimento e a senha. A operação só é realizada se a senha passada por parâmetro for igual a armazenada no cartão e o saldo da conta permitir o débito. Neste caso o nome do estabelecimento e o valor são armazenados em `relacaoDeDebitos`. O método `debitar` devolve um valor booleano indicando se a operação foi realizada.
- c) Um método `gerarFatura`, que recebe uma senha por parâmetro e imprime os nome do estabelecimento e os valores das compras do cartão. A operação só é realizada se a senha passada por parâmetro for igual a armazenada no cartão.

EXERCÍCIO 27

Através da manipulação genética, biólogos criaram uma bactéria que come lixo. Esta consome o dobro de seu peso em lixo por hora. O peso da bactéria é constante (cerca de 10 picogramas) e o lixo é totalmente metabolizado. De 3 em 3 horas cada bactéria se divide em duas. Uma bactéria vive apenas 25 horas e depois morre.

Modele a classe `Bacteria`. Ela deve representar adequadamente o estado de uma bactéria (tempo de vida, lixo metabolizado, etc). Deve ter métodos para retornar quanto lixo ela comeu, simular a passagem de uma hora (reduzindo o seu tempo de vida) e retornar o tempo de vida restante.

Crie uma classe `Colonia` que representa uma colônia de bactérias (vetor). A classe `Colonia` deve ter métodos que: simula a passagem de horas (das bactérias), retorna a quantidade de bactérias existentes e o total de lixo consumido. Faça um programa de teste que simula a passagem de N horas.

EXERCÍCIO 28

Modele e implemente uma classe chamada `ParDeDados`, composta por dois dados de seis lados e um método `rolar`. Crie uma classe `TestaParDeDados` com um método `main` que irá "lançar" (rolar) um objeto `ParDeDados` múltiplas vezes contando o número de vezes que aparece o número 6 em ambos os dados. Observe que 1 objeto `par` de dados tem 2 dados.

Pesquise pela classe `java.util.Random`.

EXERCÍCIO 29

Usando a classe `ParDeDados` da questão anterior, modele e implemente uma classe que representa um jogo chamado Pig, cujo objetivo é atingir 100 pontos antes do seu adversário. O jogo ocorre da seguinte maneira:

- a) Um jogador começa jogando um par de dados.
- b) Se os dois dados obtiverem um número diferente de 1, os pontos são acumulados no total de pontos da rodada do jogador:
 1. Se o total de pontos da rodada for menor que 20, volte para o passo (a);
 2. Se o total de pontos da rodada for maior ou igual a 20, acumule o total de pontos da rodada no total de pontos do jogador e passe o controle dos dados para o outro jogador, que começa no passo (a)
- c) Se o jogador obtiver 1 em um dos dados e NÃO estiver arriscando, este pode optar:
 1. em perder todos os pontos da rodada e passar o controle dos dados ao adversário, que começa no passo (a);
 2. em arriscar a ser o Pig. Neste caso o jogador recebe o status de `ARRISCANDO`, e volta para o passo (a) sem acumular os pontos do lançamento.
- d) Se o jogador obtiver 1 em um dos dados e JÁ estiver arriscando, perde TODOS os pontos acumulados durante o jogo (PIG) e passa o controle dos dados para o outro jogador, que começa no passo (a).

Crie as classes para representar o jogo Pig, o jogador e o programa principal (main). O jogo Pig é jogado por duas pessoas.

BOM ESTUDO!