



UNIVERSIDADE FEDERAL DO CEARÁ – UFC SOBRAL
TÉCNICAS DE PROGRAMAÇÃO – PROF. FISCHER

Lista 4: Programação Orientação a Objetos

(Pontuação: 10 pontos)

- 1) Escreva um pequeno resumo estendido (aproximadamente 5000 caracteres) que descreva e comente sobre cada um dos itens abaixo. O texto deverá ser feito com suas palavras sem cópias da internet. Ou seja, estude o conteúdo e crie seu resumo.
 - a) Classe
 - b) Objeto
 - c) Método e atributo
 - d) Encapsulamento
 - e) Visibilidade de atributos e métodos (public , private e protected)
 - f) Herança
 - g) Polimorfismo
 - h) Classes abstratas
 - i) Construtores
 - j) Get e Set
 - k) Sobrecarga de métodos
 - l) Sobrescrita de métodos
 - m) Palavras reservadas (super, this e final)
 - n) Relação de objetos: “ter”, “usar” e “ser”

- 2) A fim de representar empregados em uma firma, você deverá criar uma classe chamada Empregado. Esta classe deverá possuir as três informações a seguir:

- a. Um primeiro nome,
- b. Um sobrenome,
- c. Um salário mensal.

Sua classe deve ter um construtor que inicializa os três atributos. Forneça um método set e get para cada atributo. Se o salário mensal não for positivo, configure-o como 0.0. Crie duas instâncias da classe e exiba o salário anual de cada instância. Então dê a cada empregado um aumento de 10% e exiba novamente o salário anual de cada empregado.

- 3) Crie uma classe chamada Fatura que possa ser utilizada por uma loja de suprimentos de informática para representar uma fatura de um item vendido na loja. Uma fatura deve incluir as seguintes informações como atributos:
 - a. O número do item faturado
 - b. A descrição do item
 - c. A quantidade comprada do item

- d. O preço unitário do item.
- e. Preço total da fatura

Sua classe deve ter um construtor que inicialize os quatro atributos. Se a quantidade não for positiva, ela deve ser configurada como 0. Se o preço por item não for positivo ele deve ser configurado como 0.0. Forneça um método set e um método get para cada variável de instância. Além disso, forneça um método chamado getValorFatura que calcula o valor da fatura (isso é, multiplicar a quantidade pelo preço por item) e depois retorna o valor como um double. Crie um método que possa dar descontos em todos os itens, o porcentagem de desconto deverá ser informado para o método.

- 4) Crie um programa que tenha uma classe produto constituída de nome, preço e quantidade. Implemente um método chamado “comprarProduto”. Este método quando chamado deverá modificar uma variável do produto indicando que o produto foi comprado. Para cada produto informado escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos segundo o valor do produto conforme a tabela abaixo:
 - a. Até R\$ 100: sem desconto
 - b. De R\$ 100 a R\$ 200: 10% de desconto
 - c. De R\$ 200 a R\$ 500: 20% de desconto
 - d. Acima de R\$500: 25% de desconto
- 5) Um contato é constituído de código, nome da pessoa, endereço, e-mail, telefone e observação.
 - a. Crie 4 construtores e atribua o valores por meio dos construtores. Os construtores devem receber os seguintes parâmetros:
 - I. Código.
 - II. Código e nome.
 - III. Código, nome e e-mail.
 - IV. Telefone.
 - b. O atributo e-mail não poderá ser modificado fora da classe contato, mas poderá ser acessado por outras classes.
 - c. Faça um método para imprimir o código, nome, endereço, e-mail, telefone e observação dos contatos cadastrados.
 - d. Faça um validador do código do contato. Um código válido é maior igual a 1000 e menor igual a 9999. Não permita que seja cadastrado um código inválido para nenhum contato.
 - e. Faça um validador para o telefone. O telefone sempre deverá ter 8 dígitos. Não permita que seja cadastrado um telefone inválido para nenhum contato.
 - f. Crie um método que possa duplicar um contato. Desta forma, um novo contato terá os mesmos dados do contato que for requerido a duplicação.
 - g. Faça um método para verificar se o contato está totalmente preenchido. Ou seja, tenha valores diferentes do padrão para: código, nome, endereço, e-mail, telefone e observação.
 - h. Crie uma classe agenda que é composta de uma lista de contatos, nome da agenda e uma descrição.
 - i. Crie uma classe cliente que poderá usar uma agenda.
 - j. Faça uma classe main para testar seu sistema criado.

- 6) Crie uma implementação orientada a objetos conforme os passos abaixo:
- a) Crie uma classe chamada Ingresso que possua um atributo valor e um método toString que retorne a informação do valor do ingresso.
 - b) Crie uma classe IngressoVIP, que herda de Ingresso e possui um atributo valor Adicional. O método toString da classe IngressoVIP deve considerar que o valor do ingresso é o valor da superclasse somado ao valor Adicional do IngressoVIP.
 - c) Crie uma classe evento que é composta de uma lista ingresso.
 - d) Crie um método para possibilitar uma venda um determinado ingresso
 - e) Crie um método para computar o valor total dos ingressos vendidos
 - f) Crie um método que permita mostrar os ingressos que não foram vendidos
 - c) Crie uma classe para testar os objetos das classes Ingresso e IngressoVIP
- 7) Crie uma implementação orientada a objetos conforme os passos abaixo:
- a) A classe Empregado deve possuir dois atributos, nome e salário. O salário deve ser do tipo protected. Crie os métodos get e set para classes e o método toString.
 - b) A classe Gerente deve herdar da classe Empregado. Crie os métodos get e set para a classe e o método toString. O método toString da classe Gerente deve incluir a informação do departamento, além dos dados da superclasse. O construtor da classe deve receber por parâmetro, além das informações da superclasse, a informação do departamento.
 - c) A classe Vendedor deve herdar também da classe Empregado. Deve possuir ainda um método denominado calcularSalario. Esse método deve retornar um valor do tipo float, correspondente ao valor do salário acrescido do respectivo percentual de comissão. O construtor da classe deve receber por parâmetro, além das informações da superclasse, a informação do percentual de comissão do vendedor. O método toString da classe deve apresentar as informações de nome do empregado, salário sem comissão, salário com comissão e percentual de comissão.
 - d) Crie uma classe para testar objetos das classes implementadas.
- 8) Crie uma implementação orientada a objetos conforme os passos abaixo:
- a) A classe ContaEspecial herda da classe ContaCorrente.
 - b) Clientes que possuem conta especial possuem um limite de crédito. Dessa forma, podem fazer saques até esse valor limite, mesmo que não possuam saldo suficiente na conta.
 - c) O construtor da classe ContaEspecial deve receber como parâmetro, além dos parâmetros da superclasse, o limite que o banco disponibiliza para o cliente.
 - d) Sobrescreva o método sacar na classe ContaEspecial, de modo que o cliente possa ficar com saldo negativo até o valor de seu limite. Note que o atributo saldo da classe ContaCorrente deve ser do tipo protected para que possa ser modificado na subclasse
- 9) Crie uma classe que será chamada de Forma. A classe Forma é composta dos métodos calcularArea e imprimirNome. Crie as classes Círculo, Retângulo, Quadrado e Triângulo. Cada classe deverá ter os métodos calcularArea e imprimirNome. Porém, para cada classe deverá implementar as operações segundo cada forma geométrica. Por exemplo, o quadrado deverá calcular a área segundo o quadrado e imprimir o nome "quadrado". Utilize os recursos da orientação a objetos para modelar os requisitos acima.

- 10) Considere, como subclasse da classe Pessoa, a classe Empregado. Considere que cada instância da classe Empregado tem, para além dos atributos que caracterizam a classe Pessoa, os atributos `codigoSetor` (inteiro), `salarioBase` (vencimento base) e `imposto` (porcentagem retida dos impostos). Implemente a classe Empregado com métodos seletores e modificadores e um método `calcularSalario`. Escreva um programa de teste adequado para a classe Empregado.
- 11) Implemente a classe Operario como subclasse da classe Empregado. Um determinado operário tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo `valorProducao` (que corresponde ao valor monetário dos artigos efetivamente produzidos pelo operário) e `comissao` (que corresponde à porcentagem do `valorProducao` que será adicionado ao vencimento base do operário). Note que deverá redefinir nesta subclasse o método herdado `calcularSalario` (o salário de um operário é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.
- 12) Implemente a classe Vendedor como subclasse da classe Empregado. Um determinado vendedor tem como atributos, para além dos atributos da classe Pessoa e da classe Empregado, o atributo `valorVendas` (correspondente ao valor monetário dos artigos vendidos) e o atributo `comissao` (porcentagem do `valorVendas` que será adicionado ao vencimento base do Vendedor). Note que deverá redefinir nesta subclasse o método herdado `calcularSalario` (o salário de um vendedor é equivalente ao salário de um empregado usual acrescido da referida comissão). Escreva um programa de teste adequado para esta classe.
- 13) Crie uma classe Controle Remoto que pode controlar o volume e trocar os canais da televisão. O Controle Remoto deverá ter as seguintes funcionalidades:
- A) Ligar e desligar a TV
 - B) Aumentar ou diminuir a potência do volume de som em uma unidade de cada vez;
 - C) Aumentar e diminuir o número do canal em uma unidade trocar para um canal indicado;
 - D) Consultar o valor do volume de som e o canal selecionado;
 - E) Mudança de canal por meio de um número informado
 - F) Função Mute
- 14) Criar a classe Pessoa com as seguintes características:
- atributos: idade e dia, mês e ano de nascimento, nome da pessoa
 - métodos: o `calculaIdade()`, que recebe a data atual em dias, mês e anos e calcula e armazena no atributo `idade` a idade atual da pessoa o `informaIdade()`, que retorna o valor da idade o `informaNome()`, que retorna o nome da pessoa o `ajustaDataDeNascimento()`, que recebe dia, mês e

ano de nascimento como parâmetros e preenche nos atributos correspondentes do objeto.

- Criar dois objetos da classe Pessoa, um representando Albert Einstein (nascido em 14/3/1879) e o outro representando Isaac Newton (nascido em 4/1/1643)
- Fazer uma classe principal que cria os objetos, inicialize e mostre quais seriam as idades de Einstein e Newton caso estivessem vivos.

15) Crie uma classe Produto, essa classe deverá conter os seguintes atributos: nome do produto, preço de custo, preço de venda, margem de lucro e despesas da empresa. E o método: Cálculo da Margem de Lucros. OBS: Cálculo da Margem de Lucro: $\text{Margem de Lucro} = \{[(PV - (PC + D)) / PV]\} \times 100$. Com - PC= Preço de Custo, PV= Preço de Venda, D= Despesas. Por fim, no método main(), crie um novo objeto da classe Produto, peça para o usuário informar o nome, os preços de custo e de venda e exiba a margem de lucro em porcentagem.

16) Crie uma classe denominada Elevador para armazenar as informações de um elevador dentro de um prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- a) Inicializa : que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- b) Entra : para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- c) Sai : para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- d) Sobe : para subir um andar (não deve subir se já estiver no último andar);
- e) Desce : para descer um andar (não deve descer se já estiver no térreo); Encapsular todos os atributos da classe (criar os métodos set e get).

Somente os exercícios abaixo devem ser explicados por meio de um vídeo de no máximo 15 minutos.

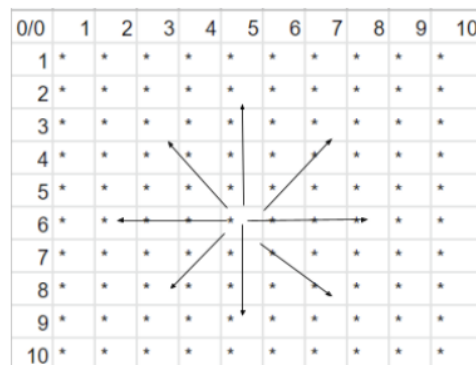
17) Escreva um programa completo para jogar o jogo da velha. Para tanto crie uma classe Jogo da Velha (OBS: apenas terminal, ainda não estudamos interface gráfica):

- a. A classe deve conter como dados privados um array bidimensional 3x3 para representar a grade do jogo.
- b. Utilize uma variável do tipo String para representar as possibilidades de ocupação de uma casa na grade (vazia, jogador 1 ou jogador 2)
- c. O construtor deve inicializar a grade como vazia
- d. Forneça um método para exibir a grade
- e. Permitir dois jogadores humanos

f. Forneça um método para jogar o jogo; todo movimento deve ocorrer em uma casa vazia; depois de cada movimento, determine se houve uma derrota ou um empate.

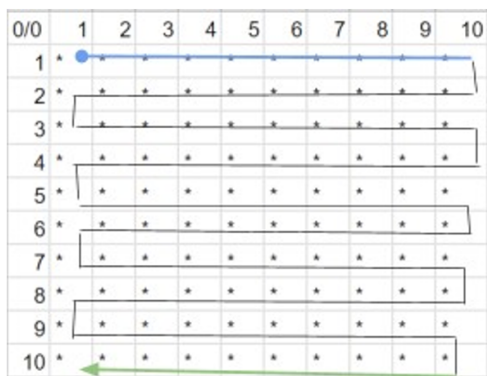
- 18) Crie uma matriz para representar uma área em que um robô pode caminhar. Sendo que o robô poderá caminhar nas direções das setas indicadas abaixo. Cada direção deverá ser mapeada por um método na classe robô. O robô deverá caminhar apenas uma casa cada vez quando um método for chamado. O robô deverá ser colocado no meio da área sempre que o programa iniciar. O tamanho da área em que o robô caminha é quadrado, mas pode ter diferentes números de linhas e colunas.

0/0	1	2	3	4	5	6	7	8	9	10
1	*	*	*	*	*	*	*	*	*	*
2	*	*	*	*	*	*	*	*	*	*
3	*	*	*	*	*	*	*	*	*	*
4	*	*	*	*	*	*	*	*	*	*
5	*	*	*	*	*	*	*	*	*	*
6	*	*	*	*	*	*	*	*	*	*
7	*	*	*	*	*	*	*	*	*	*
8	*	*	*	*	*	*	*	*	*	*
9	*	*	*	*	*	*	*	*	*	*
10	*	*	*	*	*	*	*	*	*	*

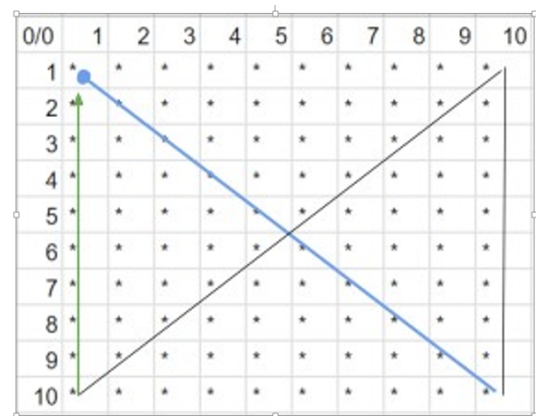


- A classe deverá ter a contagem de quantas casas o robô caminhou nas direções: norte, sul, leste, oeste, nordeste, sudeste, sudoeste e noroeste.
- O programa deverá sortear uma determinada posição X,Y e colocar uma moeda. Essa moeda deverá ser encontrada posteriormente pelo robô. Você deverá usar o símbolo “\$” para representar a moeda.
- Faça um método que ao chamar as operações para o robô caminhar encontre a moeda na área.
- Faça um método que ao chamar as operações para o robô caminhar encontre a moeda na área, com a restrição de que o mesmo método não pode ser chamado em seguida. Por exemplo, não poderia existir as seguintes chamadas: norte e norte. Mas poderia existir a seguinte combinação: norte, nordeste e norte.
- Faça um método para imprimir a área utilizada para o robô caminhar.
- Modifique os métodos utilizados para o robô caminhar na área para que cada posição da matriz que for visitada pelo robô seja marcada com o símbolo ‘@’
- Faça um método que conte as posições que o robô já passou
- Faça um método que conte as posições que o robô ainda não passou

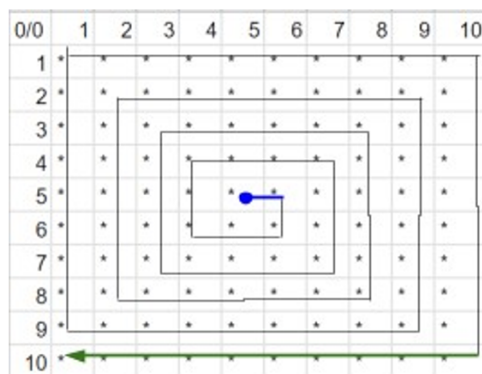
- i) Faça o robô executar as seguintes novas caminhadas descritas nos itens abaixo (um método para cada caminhada A, B e C), sempre utilize os métodos para movimento do robô. A caminhada começa no ponto (azul) e termina na flecha (verde), como demonstrado nas figuras (A, B e C) abaixo:



a)



b)



c)