UPskill – Java+.NET

Programação Orientada a Objetos – Classes e Objetos

UPskill
Digital Skills & Jobs

P.PORTO

EXERCÍCIO AUTOMÓVEL

1. Analise a classe C# disponibilizada, chamada Automóvel, para instanciar objetos que representem automóveis de combustão caracterizados pelos seguintes atributos:
   - Matrícula (Valor por omissão é "sem matrícula")
   - Marca (Valor por omissão é "sem marca")
   - Cilindrada (Valor por omissão é 0)

   Esta classe possui funcionalidades para:
   - Consultar individualmente os atributos de um automóvel;
   - Modificar individualmente os atributos de um automóvel;
   - Obter a representação textual e legível de um automóvel. Por exemplo:

   Automóvel com matrícula 24-35-AC é um Fiat e tem cilindrada de 1200 cc.
   - Determinar a diferença de cilindrada (valor relativo) entre dois automóveis
   - Verificar se a cilindrada de um automóvel é superior à de outro
   - Verificar se a cilindrada de um automóvel é superior a um determinado valor
   - Obter a quantidade de instâncias criadas
2. Compare a classe fornecida com a classe em Java fornecida com o mesmo nome.
3. Crie uma classe chamada MainAutomovel para invocar as funcionalidades da classe Automovel. Para isso:
   a) Crie uma instância da classe Automovel, designada a1, com a marca Toyota, matrícula 11-11-AA e cilindrada 1400 cc;
   b) Mostre o automóvel a1 no ecrã;
   c) Mostre apenas a matrícula do automóvel a1;
   d) Mostre a quantidade de instâncias Automovel criadas;
   e) Crie uma instância da classe Automovel, designada a2, com a marca Audi e matrícula 22-22-BB;
   f) Mostre o automóvel a2;
   g) Modifique a cilindrada do automóvel a2 para 1800 cc;
   h) Mostre novamente o automóvel a2 no ecrã;
   i) Mostre novamente a quantidade de instâncias Automovel criadas;
   j) Mostre a diferença de cilindrada (valor absoluto) entre os automóveis a1 e a2;
   k) Mostre a matrícula do automóvel que tem a maior cilindrada entre os automóveis a1 e a2;

l) Verifique se a cilindrada do automóvel a1 é superior a 2000 cc.

EXERCÍCIO DATA

1. Analise a classe Data fornecida e que foi construída com base nos seguintes requisitos:
   a) Permitir obter o estado de um objeto Data no formato "ano/mês/dia".
   b) Determinar o dia da semana de uma data, tendo em conta que:
      - O dia 1/1/1 é uma segunda-feira;
      - Um ano não bissexto avança um dia da semana (365 % 7 = 1);
      - Um ano bissexto avança 2 dias da semana (366 % 7 = 2).
      - Anos bissextos são os anos divisíveis por 4 mas não por 100, ou os divisíveis por 400.
   c) Permitir obter o estado de um objeto Data por extenso. Exemplo: terça-feira, 23 de fevereiro de 2016.
   d) Verificar se uma data é maior (mais recente) do que outra.
   e) Determinar a diferença, em dias, entre duas datas.
2. Compare a classe fornecida com a classe em Java fornecida com o mesmo nome.
3. Crie uma classe com o nome MainData para invocar todas as funcionalidades da classe Data. Para isso:
   a) Crie uma instância da classe Data com o nome data1 e que represente a data de hoje.
   b) Mostre data1 usando o formato por extenso.
   c) Crie uma outra instância da classe Data chamada data2 que guarde a data 25 de abril de 1974.
   d) Mostre data2 no formato "ano/mês/dia".
   e) Utilize o método de instância isMaior para confirmar que, de facto, data1 é mais recente do que data2.
   f) Determine o número de dias entre data1 e data2.
   g) Determine o número de dias que faltam para o Natal, usando o método que recebe, por parâmetro, o dia, o mês e o ano de uma data.
   h) Determine o dia da semana em que ocorreu o dia 25 de abril de 1974.
   i) Verifique se o ano 1974 foi bissexto, invocando o método de classe isAnoBissexto:
      - Através do objeto data2;
      - Através da classe Data.

EXERCÍCIO BANKACCOUNT

1. Implemente uma classe BankAccount que permita a gestão de contas bancárias de forma simples. Cada conta bancária deverá ter os seguintes atributos e métodos:
   - Atributos
     - AccountHolder: uma string que armazena o nome do titular da conta.
     - Balance: um valor numérico do tipo double que armazena o saldo da conta. Este saldo só pode ser modificado através de métodos específicos (não pode ser alterado diretamente).
   - Métodos:
     - Construtor
       - Deve receber dois parâmetros: o nome do titular da conta e o saldo inicial.
       - Inicializa os valores de AccountHolder e Balance.
     - Deposit
       - Recebe como parâmetro um valor do tipo double que será depositado na conta.
       - Adiciona esse valor ao saldo da conta e imprime uma mensagem informando o valor depositado e o novo saldo.
     - Withdraw
       - Recebe como parâmetro um valor do tipo double que será retirado da conta.
       - Subtrai esse valor do saldo, caso o saldo seja suficiente, e imprime uma mensagem indicando o valor retirado e o saldo restante.
       - Caso o saldo não seja suficiente, exibe uma mensagem informando que os fundos são insuficientes.
     - Transfer
       - Recebe como parâmetros outra instância de BankAccount e um valor do tipo double a ser transferido.
       - Caso o saldo seja suficiente, realiza a retirada do valor da conta atual e adiciona esse valor à conta de destino. Exibe uma mensagem indicando o valor transferido e o nome do destinatário.
       - Caso o saldo seja insuficiente, exibe uma mensagem informando que os fundos são insuficientes para realizar a transferência.

- e) PrintBalance
  - Imprime o nome do titular da conta e o saldo atual.

2. Validação de Operações Básicas da Conta
   - Crie uma instância da classe BankAccount com um nome de titular e um saldo inicial.
   - Realize as seguintes operações e verifique os resultados esperados:
     - Deposite um valor na conta e verifique se o saldo foi atualizado corretamente.
     - Retire um valor que seja menor que o saldo e verifique se o saldo foi reduzido corretamente.
     - Tente retirar um valor maior do que o saldo disponível e verifique se a mensagem de "fundos insuficientes" é exibida.
     - Transfira um valor para outra instância de BankAccount e verifique se o saldo de ambas a conta foi atualizado corretamente.
     - Imprima o saldo para garantir que o saldo final corresponde às operações realizadas.