ADS – IFPB – Campus Monteiro

LISTA SEMANAL – PROGRAMAÇÃO II - PROGRAMAÇÃO ORIENTADA A OBJETOS  
Prof. Cleyton Caetano de Souza

**Observação**: essa lista pode ser feita em dupla.

Semana 12

**Datas Válidas**

1. Considere a classe MinhaData, da lista semanal 11, faça o construtor da classe MinhaData lançar uma exceção (**Exception**) caso a data que se está tentando criar não exista no calendário (ex. 30/02/2017). Crie um main que requisite ao usuário que informe a data na forma “dia/mes/ano” (ex. 30/01/1999) e, em seguida, tente criar um objeto MinhaData e informe se a data fornecida pelo usuário é válida ou não. Enquanto o usuário não digitar uma data válida, o programa deve ler novamente as informações da data, na forma “dia/mes/ano”. **Sugestão:** pesquise formas que já existem para validar a data.

**Impressora OO**

2. **[Questões do Monitor - Maycon]** Crie a classe Impressora com os atributos: quantidade de páginas (do tipo inteiro) e carga do toner (do tipo float). Inicie esses atributos via construtor (a quantidade de páginas começa com 0 e a quantidade de toner começa com 100). Adicione um método chamado imprimir que recebe como parâmetro de entrada uma quantidade de páginas e retorna a quantidade de páginas que foram impressas. O método imprimir reduz a quantidade de páginas da impressora (baseado em quantas páginas vão ser impressas) e também reduz a quantidade de toner (a impressora gasta 0.35 de tinta a cada página impressa). Caso, no meio da impressão, não haja mais papel suficiente para completar a execução do método, ele deve lançar uma exceção chamada **FaltouPapelException**. Essa exceção deve ter um atributo para guardar a quantidade de páginas que ainda não foram impressas do último trabalho de impressão. Caso a tinta seja insuficiente para continuar a impressão, o método lança uma **FaltouTintaException**.

3. **[Questões do Monitor - Maycon]** Crie um programa onde você irá mostrar, ao usuário, as seguintes opções: 1 – imprimir, 2 – carregar papel na impressora, 3 – trocar o toner da impressora e 4 - sair. Ao escolher a opção imprimir, você deverá ler a quantidade de páginas a serem impressas e usar a impressora para imprimi-las. Ao escolher a opção de carregar papel, você deve ler qual a quantidade de páginas que o usuário quer acrescentar a impressora. Ao escolher a opção recarregar toner, o toner da impressora volta a ter 100 de carga. Ao escolher a opção sair, informe a quantidade total de páginas que foram impressas, durante a execução do programa.

**Radar OO**

4. Crie um programa que simule um equipamento eletrônico de monitoramento de velocidade de veículos. Inicialmente, teremos uma classe Veiculo, ela possuirá os seguintes atributos placa (String), velocidadeAtual (float), nomeDoProprietario (String). O programa terá uma classe chamada Radar que possuíra o método **verificarVelocidade**. A classe Radar possui um atributo, inicializado via construtor, que corresponde a velocidade máxima permitida. O método verificarVelocidade recebe um veículo como entrada e verifica se sua velocidade está acima do limite máximo permitido pelo radar. Se estiver, o método deve informar que está acima da velocidade permitida lançando uma **VelocidadeMaximaNaoRespeitadaException**. Se a velocidade do veículo for menor que 50% da velocidade máxima, deve ser lançada uma **VelocidadeMinimaNaoRespeitadaException**. Essas exceções devem ser subclasses de Exception e possuir mensagens personalizadas explicando o erro. Caso o veículo esteja dentro do limite permitido, o método não faz nada (esse método não tem retorno).

5. Crie uma entidade chamada Multa que possui como atributos: um Veículo e float representando o valor da multa a ser paga. Adicione um construtor que recebe o veículo como entrada e o valor da multa a ser pago. **Adicione as exceções da questão anterior um atributo do tipo Multa e no método verificar velocidade adicione a Multa à exceção antes dela ser lançada**. O valor da multa respeita as seguintes regras:

* Se o excesso de velocidade é até 20% do limite permitido a multa deve ser de **R$ 85,13**
* Se o excesso de velocidade é entre 20% até 50% do limite permitido a multa deve ser de **R$ 127,53**
* Se o excesso de velocidade é acima de 50% do limite permitido a multa deve ser de **R$ 574,62.**
* **A multa por dirigir abaixo do limite mínimo tem valor fixo e é de R$ 87,33**

6. Por meio da questão anterior, crie um main que simule o uso de um Radar na rodovia. Para isso, crie uma lista com 100 carros com velocidades aleatórias entre 40 km/h e 150 km/h. Crie um radar cuja velocidade máxima deve ser 90 km/h. O Radar deve receber todos os carros e informar se ele foi multado ou não e o valor da multa, quando for o caso. Ao fim, o programa deve informar quantas multas o radar “deu” e o valor total delas.

7. Escreva um programa para preencher valores de um vetor de inteiros com 10 posições. O usuário irá informar os valores a serem inseridos e suas respectivas posições no array. Utilize a classe JOptionPane para Entrada e Saída de dados: para a leitura, use o método showInputDialog [**[1]**](https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html#showInputDialog(java.lang.Object)); para impressão, use o showMessageDialog [**[2]**](https://docs.oracle.com/javase/7/docs/api/javax/swing/JOptionPane.html#showMessageDialog(java.awt.Component,%20java.lang.Object)) **(pesquise sobre como usar essa classe e esses métodos)**. O programa deve tratar as exceções **IndexOutOfBoundsException** quando for informada uma posição inexistente do vetor e **NumberFormatException** quando o valor informado não for um número. Quando todas as posições estiverem com valores diferentes de zero, encerre o programa e exiba a soma dos números guardados no array.