Lista de Exercícios 04: Critérios de Testes Funcionais

1. **Tabela de decisão.** Cálculo da hipoteca. Considere os seguintes requisitos:

   - R1- O sistema deve receber três valores como entrada: gênero (1 → feminino e 0 → masculino), idade ([21, 70]) e salário ([0-10000]). Como saída, o sistema deve calcular o valor máximo da hipoteca para essa pessoa.
   - R2- O valor máximo da hipoteca é calculado pela multiplicação do valor do salário com um fator (ver tabela).
   - R3- Mensagens de erro especificas devem ser geradas para valores inválidos de idade e salário.
   - R4- O fator para calcular a hipoteca (R2) e definido pela tabela a seguir:

Homem	Fator	Mulher	Fator
21-40 anos	**80**	**21-43 anos**	**60**
41-70 anos	**40**	**44-70 anos**	**30**

Implemente uma classe Java que calcula o valor máximo da hipoteca e casos de teste JUnit derivados da **tabela de decisão**. Elabore uma planilha que ilustra a tabela de decisão criada.

2. **Tabela de decisão.** A classe Calculadora e usada para calcular o salário dos empregados de acordo com as seguintes regras:
   - Se o tipo de empregado e "Assalariado40H" seu salário e 4.000.
   - Se o tipo de empregado e "Assalariado20H" seu salário e 1.500.
   - Se o tipo de empregado e "Horista" seu salário e o número de horas trabalhadas vezes 15.
     - Caso o horista trabalhe exatamente 40 horas nenhuma pendência e gerada (atributo pendência da classe *Salário*).
     - Caso o horista trabalhe menos de 40 horas, a pendência "relatório de ausência" e gerada.
     - Caso o horista trabalhe mais de 40 horas, a pendência "autorização de hora extra" e gerada.

Elabore uma tabela de decisão para a descrição apresentada. Derive casos de teste da tabela e implemente os casos de teste em JUnit, considerando as classes a seguir.

Salario.java

```
public class Salario {
    int valorSalario;
    String pendencia;

public String getPendencia() {
    return pendencia;
}
public int getValorSalario() {
    return valorSalario;
}
}
```

Calculadora.java

```
public class Calculadora {

public Salario calcularSalario(String tipoEmpregado, int horasTrabalhadas) {
    //todo
    return null;
    }
}
```

3. **Partição em Classes de Equivalências.** Uma universidade considera aprovado o aluno que obtém nota maior ou igual a 7. Porém, com nota maior ou igual a 4 ele terá direito a uma prova de recuperação, mas com nota inferior a 4 estará reprovado. Considerando apenas valores para uma nota válida (entre 0 e 10), qual alternativa apresenta uma nota para cada partição equivalente?

   a) 2.5, 6.0, 8.9
   b) 3.0, 5.5, 6.0
   c) 3.5, 7.8, 9.0
   d) 4.5, 5.5, 9.0

Implemente uma classe Java que calcula os valores de cada classe de equivalência e casos de teste JUnit.

4. **Partição em Classes de Equivalências**. Um sistema solicita a nota bimestral de um aluno. As notas permitidas são entre 0 e 10. Notas fora desse intervalo são consideradas "notas inválidas". Considerando o critério Partição por Classes de Equivalência, identifique as possíveis classes existentes e implemente em JUnit um caso de teste para cada classe.
   Elabore uma tabela com os casos de teste para a descrição apresentada.

5. **Partição em Classes de Equivalências.** Considerando a idade do eleitor, o sistema deve informar se ele não pode votar, é obrigado a votar ou se o voto é facultativo. Para ser obrigado a votar o eleitor deve ter idade entre 18 e 70 anos. É proibido eleitores menores de 16 anos a votarem. E é facultativo os eleitores com menos de 16 ou mais de 70 anos.
   Elabore uma tabela com os casos de teste para a descrição apresentada.

6. **Análise de Valor Limite.** Considerando os exercícios 4 e 5, complemente os casos de teste usando o critério de análise do valor limite. Implemente em JUnit e complemente as tabelas criadas para os casos de teste.