Técnicas de Conceção de Testes

- **Sugestão**
  - Usem tabelas já preparadas (exemplos)
    - **Para apresentar as partições equiv. e valores fronteira**

Condições de entrada	Partição válida	Partição inválida	Fronteira válida	Fronteira inválida
Nome (dimensão)	2-60	< 2	2	1
Idade (valor int)	>= 18	< 18	18	17
…	…	…	…	…

    - **Para apresentar os casos de teste**

Caso de Teste	Descrição	Resultado esperado	Condições cobertas
1	Nome = Luis Pereira Idade = 22, …	Próximo form	PV1, PV2, PV3, FV2, …
2			
…			

Fundamentos de Teste de Software

O código fornecido corresponde a um programa para realizar 2 operações matemáticas.

Foi desenvolvido com base nas seguintes especificações:

- *int fatorial(int n)* - deve retornar o fatorial no número n. Apenas deverá aceitar números iguais ou superiores a zero.
  - O fatorial de '0' é 1
  - O fatorial de n (se n ≥ 1) é dado por: 1×2×...×n
  - Para as entradas inválidas o programa deve retornar o valor -1
- *boolean ePrimo(int n)* - verifica se um número é primo. Retorna um valor booleano que confirma se o número é primo ou não. Um número é primo se apenas é divisível por 1 e por ele próprio. Apenas deverão ser aceites números inteiros superiores a 1. Todos os que não satisfizerem esta condição devem ser tratados como falsos primos

Considerando as especificações e o código fornecido, escreva o conjunto de casos de teste que realizaria (apresente todos os passos realizados).


```java
public class Operacoes {

    /**
     * Método para calcular o fatorial de um número.
     * @param n número inteiro.
     * @return <code>integer</code> indicando o resultado do cálculo do fatorial.
     */
    public int fatorial(int n){
        int result = 1;

        if (n < 0)
            result = -1;
        else
            for (int i = 1; i <= n; i++)
                result = result * i;
        return result;
    }

    /**
     * Método para determinar se um número é primo ou não.
     * @param n número inteiro para verificar
     * @return <code>boolean</code> verdadeiro ou falso consoante 'n' é primo ou não
     */
    public boolean ePrimo(int n){
        if (n < 2)
            return false;

        for (int i = 2; i < n; i++)
        {
            if (n % i == 0)
                return false;
        }
        return true;
    }

}
```

Fundamentos de Teste de Software

- **Avaliações**
  - Dia 19 de abril, Trabalho prático 1
  - Dia 26 de abril, Prova escrita
  - Dia 24 de maio, Trabalho prático 2

Técnicas de Conceção de Testes

- **Técnicas de conceção de Testes**
  - Técnicas Baseadas na Estrutura ou **Caixa-Branca**

Técnicas de Conceção de Testes

- **Técnicas de conceção de Testes**
  - Técnicas Baseadas na Estrutura ou **Caixa-Branca**
  - Objetivos de aprendizagem:
    - **Descrever o conceito e valor da cobertura de código**
    - **Explicar os conceitos de cobertura de instruções ou decisões**
    - **Escrever casos de teste a partir de fluxos de controlo utilizando as técnicas de conceção de testes por instrução ou decisão**
    - **Avaliar a cobertura de instruções e decisões face ao cumprimento integral de critérios de saída definidos.**

Técnicas de Conceção de Testes

- **Técnicas de conceção de Testes**
  - Técnicas Baseadas na Estrutura ou **Caixa-Branca**
  - Objetivo
    - **determinar defeitos da estrutura interna do programa, através de testes que exercitem suficientemente os caminhos de execução possíveis**
  - Funcionamento
    - **baseia-se no conhecimento da estrutura interna do programa**
    - **são testados os caminhos lógicos estabelecendo casos de teste que põem à prova condições, definições e variáveis**

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Baseiam-se na estrutura do software ou sistema:
    - **Ao nível do componente:**
      - a estrutura de um componente de software, ou seja, as suas instruções, decisões, ramos ou caminhos distintos
    - **Ao nível da integração:**
      - a estrutura pode ser um diagrama no qual módulos invocam outros módulos
    - **Ao nível do sistema:**
      - a estrutura pode ser um menu estruturado, um processo de negócio ou a estrutura de uma página web.

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Testes de Instrução e sua Cobertura
  - Testes de Decisão e sua Cobertura
  - Outras Técnicas baseadas na Estrutura.

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Testes de Instrução e sua Cobertura
    - **Os casos de teste são concebidos para executar instruções específicas de forma a aumentar a cobertura de instruções**
    - **Em testes de componentes, a cobertura de instruções é a avaliação da percentagem de instruções executáveis que tenham sido executadas por um conjunto de casos de teste**
    - **A cobertura de instruções é determinada pelo número de instruções executáveis cobertas por casos de teste, divididas pelo número total de instruções executáveis do código em teste.**

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Testes de Decisão e sua Cobertura
    - **Os casos de teste são concebidos de modo a executar os resultados de decisões específicas**
    - **A cobertura de decisões, relativamente aos testes de ramos, é a avaliação da percentagem de resultados de decisões (p. ex. as opções Verdadeiro e Falso de uma instrução SE), que tenham sido já executadas por um conjunto de casos de teste**
    - **Os ramos são originários dos pontos de decisão no código e mostram a transferência do controlo para diferentes localizações do código.**

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Testes de Decisão e sua Cobertura
    - **A cobertura de decisões é determinada pelo número de todos os resultados de decisões cobertas por casos de teste, dividido pelo número de todos os resultados de decisões possíveis no código em teste**
    - **Os testes de decisão são uma forma de testes de fluxo de controlo pois seguem um fluxo de controlo específico através dos pontos de decisão**
    - **A cobertura de decisões é mais eficaz do que a cobertura de instruções:**
      - 100% de cobertura de decisões assegura 100% de cobertura de instruções, mas não vice-versa.

Técnicas de Conceção de Testes

- **Baseadas na Estrutura ou Caixa-Branca**
  - Outras Técnicas baseadas na Estrutura
    - **Existem níveis mais elevados de cobertura estrutural além da cobertura de decisões, por exemplo:**
      - cobertura de condições
      - cobertura de múltiplas condições.
    - **O conceito de cobertura também pode ser aplicado a outros níveis de teste, por exemplo:**
      - nível da integração, a percentagem de módulos, componentes ou classes que têm sido executadas por um conjunto de casos de teste pode ser expressa como cobertura de componente, módulo ou classe.
    - **O recurso a ferramentas é muito útil para os testes de código estruturais.**

Técnicas de Conceção de Testes

- **Técnicas de conceção de Testes**
  - Técnicas baseadas na Experiência
  - Objetivos de aprendizagem:
    - **Recordar as razões de conceber casos de teste com base na intuição, experiência e conhecimento sobre defeitos comuns**
    - **Comparar as técnicas baseadas na experiência com as técnicas baseadas nas especificações.**

Técnicas de Conceção de Testes

- **Técnicas de conceção de Testes**
  - Técnicas baseadas na Experiência
    - **São testes baseados na experiência com aplicações e tecnologias similares e na intuição do testador**
    - **Podem reforçar as técnicas sistemáticas**
    - **São úteis na identificação de testes especiais menos fáceis de capturar pelas técnicas formais**
    - **No entanto, esta técnica poderá produzir variados graus de eficácia, dependendo da experiência do testador**
    - **Uma das técnicas baseada na experiência mais utilizada é a de antecipar erros**
      - os testadores antecipam os defeitos com base na sua experiência
      - Enumeram uma lista de possíveis defeitos e concebem os testes para atacar esses defeitos

Técnicas de Conceção de Testes

- **Usando ferramentas para executar e gerir os testes**
  - Técnicas baseadas na Estrutura ou Caixa-Branca
    - **Testes de Instrução e sua Cobertura**
  - Desenvolvimento dirigido por testes