Introdução a Testes de Software
– Análise do Valor Limite –

Esses slides usam o conteúdo do livro Software Testing: From Theory to Practice, de Maurício Aniche (Universidade Técnica de Delft), disponível online no endereço sttp.site.

Análise do Valor Limite

- Erros similares a usar “>” quando deveríamos usar “>=”, por exemplo, são comuns.
- Programas com esse tipo de defeito tendem a quase sempre funcionar corretamente.
- Mas falham quando a entrada exercita o(s) valor(es) limite(s) da condição.
  - Fronteira da condição

Análise do Valor Limite

- Análise do Valor Limite complementa o critério de Particionamento por Equivalência, exigindo casos de teste nos limites (fronteiras) de cada partição.
  - Teste de fronteiras (*Boundary Testing*)

Análise do Valor Limite

- Partições tem fronteiras com outras partições
- Se formos fazendo pequenas mudanças em um valor de entrada de uma partição (ex.: somando 1), em algum ponto essa entrada pertencerá a outra partição.
  - A esse ponto chamamos de fronteira
- A análise do valor limite trata da definição de casos de teste para testar se o programa funciona corretamente com entradas próximas das fronteiras.

Exemplo – Pontos de Jogador

- Dados os pontos de um jogador e o número de suas vidas restantes, o programa faz o seguinte:
  - Se o jogador tiver menos que 50 pontos, outros 50 pontos são adicionados aos pontos atuais do jogador;
  - Se o jogador tiver 50 pontos ou mais, então:
    - se o número de vidas for maior ou igual a 3, o número de pontos do jogador é triplicado;
    - caso contrário, 30 pontos são adicionados aos pontos atuais.

Exemplo - Pontos de Jogador

```
public class PontosJogador {

  public int calcularPontosTotais(int pontosAtuais, int vidasRestantes) {
    if (pontosAtuais < 50)
      return pontosAtuais+50;

    return vidasRestantes < 3 ? pontosAtuais+30 : pontosAtuais*3;
  }

}
```

Exemplo – Pontos de Jogador

- Partições
  - **1. Poucos pontos:**
    - pontosAtuais < 50
    - Caso de teste
      - pontosAtuais = 30, vidasRestantes = 5, saída = 30 + 50
  - **2. Muitos pontos, mas poucas vidas:**
    - pontosAtuais >= 50 e vidasRestantes < 3
    - Caso de teste
      - pontosAtuais = 300, vidasRestantes = 1, saída = 300 + 30
  - **3. Muitos pontos e muitas vidas:**
    - pontosAtuais >= 50 e vidasRestantes >= 3
    - Caso de teste
      - pontosAtuais = 500, vidasRestantes = 10, saída = 500 * 3

Exemplo - Pontos de Jogador

Implementação de uma suíte de teste que não leva em conta as fronteiras

```
public class PontosJogadorTest {

  private final PontosJogador pj = new PontosJogador();

  @Test
  void poucosPontos() {
    assertEquals(30+50, pj.calcularPontosTotais(30, 5));
  }

  @Test
  void muitosPontosPoucasVidas() {
    assertEquals(300+30, pj.calcularPontosTotais(300, 1));
  }

  @Test
  void muitosPontosMuitasVidas() {
    assertEquals(500*3, pj.calcularPontosTotais(500, 10));
  }
}
```

Exemplo – Pontos de Jogador

- Fronteiras
  - **F1**
    - Se pontosAtuais for menor que 50, ele pertence a partição 1. Se pontosAtuais for igual ou maior que 50, ele pertence as partições 2 ou 3. Então, temos a seguinte fronteira: quando pontosAtuais muda de 49 para 50, sua partição também muda.
    - Caso de teste
      - F1.1: entrada = {pontosAtuais = 49, vidasRestantes = 5}, saída = {49+50}
      - F1.2: entrada = {pontosAtuais = 50, vidasRestantes = 5}, saída = {50*3}
  - **F2**
    - Dado que pontosAtuais é maior que 50, se vidasRestantes for menor que 3, ele pertence à partição 2, caso contrário, pertence à partição 3.
    - Caso de teste
      - F2.1: entrada = {pontosAtuais = 500, vidasRestantes = 2}, saída = {500+30}
      - F2.2: entrada = {pontosAtuais = 500, vidasRestantes = 3}, saída = {500*3}

Exemplo - Pontos de Jogador

Casos de teste que verificam as fronteiras

```
@Test
void entrePoucosEMuitosPontos() {
  assertEquals(49+50, pj.calcularPontosTotais(49, 5));
  assertEquals(50*3, pj.calcularPontosTotais(50, 5));
}

@Test
void entrePoucasEMuitasVidas() {
  assertEquals(500+30, pj.calcularPontosTotais(500, 2));
  assertEquals(500*3, pj.calcularPontosTotais(500, 3));
}
```

Exemplo: Barras de Chocolate

- Um pacote pode conter um certo número de barras de chocolate, que podem ser pequenas (1 quilo cada) ou grandes (5 quilos cada)
- Assumindo que o pacote é sempre preenchido com o maior número possível de barras grande, retorne o número de barras pequenas necessárias para encher a caixa. E retorne -1 se não é possível encher o pacote completamente.
- A entrada do programa é: número de barras pequenas disponíveis, número de barras grandes disponíveis, e a quantidade de quilos que o pacote suporta.

Exemplo: Barras de Chocolate

Uma possível implementação para esse programa:

```
public class ChocolateBars {

    public static final int CANNOT_PACK_BAG = -1;

    public int calculate(int small, int big, int total) {
        int maxBigBoxes = total / 5;
        int bigBoxesWeCanUse = Math.min(maxBigBoxes, big);
        total -= (bigBoxesWeCanUse * 5);

        if(small <= total)
            return CANNOT_PACK_BAG;
        return total;

    }
}
```

Exemplo: Barras de Chocolate

Casos de Teste

- **Apenas barras pequenas:** small = 4, big = 2, total = 3
- **Apenas barras grandes:** small = 5, big = 3, total = 10
- **Barras pequenas e barras grandes:** small = 5, big = 3, total = 17
- **Não há barras suficientes:** small = 1, big = 1, total = 10
- **Número negativo nos parâmetros:** small = -1, big = -1, total = -1

Exemplo: Barras de Chocolate

- Mas se fornecermos a seguinte entrada?
  - small = 2, big = 3, total = 17
- Ela pertence a partição **Barras pequenas e barras grandes**
  - O resultado deve ser 2 (duas barras pequenas)

Exemplo: Barras de Chocolate

- Mas se fornecermos a seguinte entrada?
  - small = 2, big = 3, total = 17
- Ela pertence a partição **Barras pequenas e barras grandes**
  - O resultado deve ser 2 (duas barras pequenas)
  - Mas o programa retorna -1

- Nosso conjunto de testes não foi suficiente para detectá-lo

Exemplo: Barras de Chocolate

small = 2, big = 3, total = 17

Partiçoes

- **Apenas barras pequenas:** small = 4, big = 2, total = 3
- **Apenas barras grandes:** small = 5, big = 3, total = 10
- **Barras pequenas e barras grandes:** small = 5, big = 3, total = 17
- **Não há barras suficientes:** small = 1, big = 1, total = 10
- **Número negativo nos parâmetros:** small = -1, big = -1, total = -1

Exemplo: Barras de Chocolate

Uma possível implementação para esse programa:

```
public class ChocolateBars {

    public static final int CANNOT_PACK_BAG = -1;

    public int calculate(int small, int big, int total) {
        int maxBigBoxes = total / 5;
        int bigBoxesWeCanUse = Math.min(maxBigBoxes, big);
        total -= (bigBoxesWeCanUse * 5);

        if(small <= total)
            return CANNOT_PACK_BAG;
        return total;

    }
}
```

Exemplo: Barras de Chocolate

Uma possível implementação para esse programa:

```
public class ChocolateBars {

    public static final int CANNOT_PACK_BAG = -1;

    public int calculate(int small, int big, int total) {
        int maxBigBoxes = total / 5;
        int bigBoxesWeCanUse = Math.min(maxBigBoxes, big);
        total -= (bigBoxesWeCanUse * 5);

        if(small < total)
            return CANNOT_PACK_BAG;
        return total;

    }
}
```

Barras de Chocolate - Fronteiras

small = 0, big = 3, total = 17, = -1

small = 1, big = 3, total = 17, = -1

small = 2, big = 3, total = 17, = 2

small = 3, big = 3, total = 17, = 3

Barras de Chocolate - Fronteiras

small = 0, big = 3, total = 17, = -1

small = 1, big = 3, total = 17, = -1

Não há barras suficientes

---

small = 2, big = 3, total = 17, = 2

small = 3, big = 3, total = 17, = 2

Barras pequenas e barras grandes

Ex.: small = 5, big = 3, total = 17

Barras de Chocolate - Fronteiras

small = 2, big = 3, total = 14, = -1

small = 3, big = 3, total = 14, = -1

Não há barras suficientes

---

small = 4, big = 3, total = 14, = 4

small = 5, big = 3, total = 14, = 4

Barras pequenas e barras grandes

Ex.: small = 5, big = 3, total = 17

Barras de Chocolate - Fronteiras

small = 1, big = 1, total = 6, = 1 Barras pequenas e barras grandes

---

small = 1, big = 1, total = 7, = -1

small = 1, big = 1, total = 8, = -1

Não há barras suficientes
Ex: small = 1, big = 1, total = 10

Barras de Chocolate - Fronteiras

small = 5, big = 1, total = 10, = 5 Barras pequenas e barras grandes

---

small = 5, big = 2, total = 10, = 0

small = 5, big = 3, total = 10, = 0

Apenas barras grandes
Ex: small = 5, big = 3, total = 10

Barras de Chocolate - Fronteiras

small = 2, big = 2, total = 3, = -1 Não há barras suficientes

---

small = 3, big = 2, total = 3, = 3

small = 4, big = 2, total = 3, = 3

Apenas barras pequenas

Ex.: small = 4, big = 2, total = 3