Verificação e Validação de Software
Prof. Bernardo Copstein

Geração de Casos de Teste para Teste de Unidade

Exercício para entregar

Enunciado do problema

Em uma diminuta ilha tropical a gasolina vendida nos postos é resultado de uma mistura de 3 componentes: 5% de aditivo, 25% de álcool e 70% de gasolina pura. A preparação do produto é feita por encomenda dos postos em um único centro de distribuição. Esse centro possui estoques adequados ao número de veículos da ilha. A figura 1 apresenta os tanques disponíveis nesse centro.

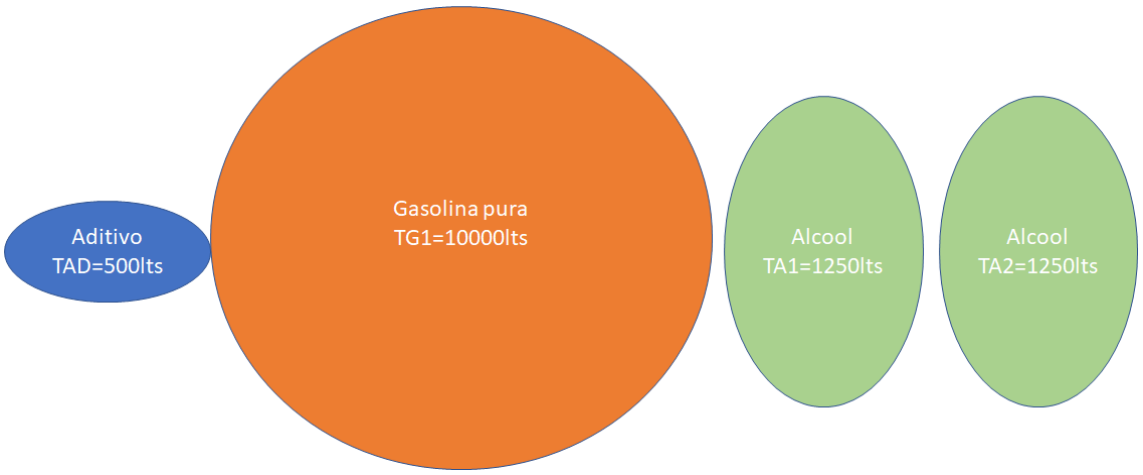

Figura 1 – Tanques do centro de distribuição de combustível.

Existem dois tipos de postos na ilha. Os postos COMUNS, que atendem a população em geral e os postos ESTRATEGICOS que atendem apenas ambulâncias e veículos da polícia.

A produção da mistura que será entregue nos postos deve-se obedecer aos percentuais já definidos e considerar as seguintes regras:

1. Se houver falta de qualquer um dos componentes na quantidade adequada a encomenda não pode ser entregue.
2. Os tanques de álcool devem ter sempre a mesma quantidade de combustível de maneira a manter o equilíbrio da estrutura devido a forma como foram construídos. Isso vale tanto para o armazenamento como para a retirada.
3. Como os tanques são pequenos a ilha mantém um sistema para controlar os estoques. Quando todos os tanques estiverem com pelo menos 50% da capacidade o sistema opera em modo NORMAL e as encomendas são entregues normalmente para qualquer tipo de posto. Se o volume armazenado em qualquer um dos tanques cair abaixo de 50% o sistema passa a operar em modo SOBRAVISO. Neste modo o sistema só entrega 50% do que é solicitado pelos postos COMUNS e o total solicitado pelos ESTRATEGICOS. Caso o volume em qualquer um dos tanques caia abaixo de 25%, então o sistema passa a operar em modo de EMERGÊNCIA e as encomendas dos postos COMUNS deixam de ser atendidas e as dos ESTRATÉGICOS são atendidas em 50%.

Estrutura da classe

A estrutura da classe “CentroDistribuicao” pode ser vista na figura 2.

```
public class CentroDistribuicao {
   public enum SITUACAO { NORMAL, SOBRAVISO, EMERGENCIA }
   public enum TIPOPOSTO { COMUM, ESTRATEGICO }

   public static final int MAX_ADITIVO = 500;
   public static final int MAX_ALCOOL = 2500;
   public static final int MAX_GASOLINA = 10000;

   public CentroDistribuicao (int tAditivo, int tGasolina, int tAlcool1, int tAlcool2) { ... }

   public void defineSituacao(){ ... }

   public SITUACAO getSituacao(){ ... }

   public int gettGasolina(){ ... }

   public int gettAditivo(){ ... }

   public int gettAlcool1(){ ... }

   public int gettAlcool2(){ ... }

   public int recebeAditivo(int qtdade) { ... }

   public int recebeGasolina(int qtdade) { ... }

   public int recebeAlcool(int qtdade) { ... }

   public int[] encomendaCombustivel(int qtdade, TIPOPOSTO tipoPosto) { ... }
}
```

Figura 2 – Estrutura da classe “CentroDistribuicao”

O método construtor recebe as quantidades iniciais de gasolina nos tanques e ajusta a “situação” de acordo. Caso algum dos parâmetros tenha valor inválido o método deve gerar uma ILLEGAL_ARGUMENT_EXCEPTION (isso vale também para quantidades iniciais de álcool que devem ser iguais). O método “defineSituacao” ajusta a situação de acordo com as regras. Ele deve ser chamado tanto pelos métodos que sinalizam a chegada de componentes no centro de distribuição quanto pelo método “encomendaCombustivel” que sinaliza o fornecimento de combustível para um posto. Os métodos “recebeAditivo”, “recebeGasolina” e “recebeAlcool” são usados quando o centro de distribuição recebe carga dos componentes. Todos recebem por parâmetro a quantidade do componente (aditivo, gasolina ou álcool) recebida e retornam à quantidade que puderam armazenar devido a limitação do tamanho dos tanques e de quanto ainda tinham armazenado. Devem retornar “-1” caso a quantidade recebida por parâmetro seja inválida. O método “encomendaCombustivel” é usado quando o centro de distribuição recebe o pedido de um posto. Este método recebe por parâmetro a quantidade solicitada pelo posto e o tipo do posto. Se o pedido puder ser atendido, o método retorna um arranjo com a quantidade de combustível remanescente em cada tanque, depois do pedido atendido. As quantidades devem ser retornadas pela ordem: aditivo, gasolina, álcool T1 e álcool T2. A primeira posição do

arranjo é usada também para indicar códigos de erro. No caso de ser recebido um valor inválido por parâmetro deve-se retornar “-7” na primeira posição do arranjo, se o pedido não puder ser atendido em função da “situação” retorna-se “-14” e, caso não haja combustível suficiente para completar a mistura, retorna-se “-21”. Por simplicidade trabalha-se apenas com números inteiros. Na hora de fazer os cálculos multiplique os valores por 100. Depois de feitos os cálculos dividam por 100 novamente e despreze a parte fracionária;

Tarefas

As tarefas que seguem referem-se aos métodos listados no item “Estrutura da Classe”. Ao final deverá ser entregue um relatório contendo:

1. Projeto dos casos de teste. Para cada uma das técnicas utilizadas presentar uma tabela com os casos de teste e resultados esperados para cada teste. Detalhar a aplicação das estratégias (ex: partições definidas, limites explorados, grafos etc.).
2. Código da classe driver (explorando testes parametrizados quando possível)
3. Código da classe alvo
4. Defeitos encontrados/corrigidos
5. Análise de cobertura de código usando o “Code Coverage”.
6. Casos de teste adicionais visando cobertura de blocos e de decisão
7. Código dos métodos de teste adicionais
8. Eventuais defeitos adicionais encontrados
9. Análise das técnicas e ferramentas utilizadas para a melhoria do código gerado

Atenção: cada grupo deverá desenvolver seu código da classe alvo e pode usar seu próprio conjunto de casos de teste para verificar a implementação. Antes de dar sequência no trabalho, porém, deverá trocar sua implementação com a de outro grupo. O objetivo é que os testes – e o relatório – sejam feitos a partir de uma implementação diferente da sua. **Essa troca deverá ser feita na aula do dia 29/09.**

Apresentação

A apresentação deverá enfatizar dois aspectos:

- A análise do software verificado

- O uso de recursos do JUnit não explorados em aula