Universidade de São Paulo - Escola de Artes, Ciências e Humanidades
ACH2028 – Qualidade de Software
Prof. Marcelo Medeiros Eler

Atividade 03 - Teste Funcional ou Caixa preta

Esta atividade consiste em ler a especificação de um sistema de software e criar casos de teste com base nos critérios particionamento em classes de equivalência e análise de valor-limite da técnica de teste funcional ou caixa preta. O software descrito a seguir foi adaptado de um exercício proposto na disciplina CSC326 – Software Engineering da NC State University.

Máquina de fazer café

1 - Descrição geral

Uma máquina de fazer café foi instalada na universidade para ser utilizada pelos alunos de graduação e pós-graduação. Apesar de se chamar máquina de café, a máquina pode fazer qualquer receita que combine café, leite, chocolate e açúcar. A máquina possui diversas operações disponíveis e será acessada por meio de uma interface que possui local para inserir valores em moeda ou notas, botões para selecionar a bebida, local para receber a bebida pronta, e local para receber o troco (se houver). Entretanto, o software descrito a seguir refere-se apenas ao código de negócio da aplicação, e não à interface já implementada. A interface da máquina deverá se comunicar com as operações públicas disponibilizadas pelo software de controle da máquina de café.

2 - Operações disponíveis para ser utilizada pela interface da máquina de café

Criar Receita

Uma receita consiste de um nome, um preço, unidades de café, unidades de leite, unidades de açúcar, e unidades de chocolate. As unidades podem ter valor 0, mas pelo menos um ingrediente deve ter unidade maior do que 0. O preço não pode ser 0 e é expresso por um número inteiro que representa quantos centavos custa a bebida. O nome da receita não pode ser vazio. Deve haver uma forma de saber se a receita foi criada com sucesso ou se houve problema com os valores dos ingredientes.

Inserir Receitas

Somente três receitas podem ser cadastradas na máquina de café. Cada receita tem que ter um nome único na lista de receitas e não pode haver duas receitas com as mesmas quantidades de ingredientes. O preço e as unidades da receita devem ser números inteiros. Deve haver uma forma de saber se uma receita foi inserida com sucesso ou se houve algum problema, como, por exemplo, valores insuficientes, receita repetida ou limite de receitas excedido, por exemplo.

Remover uma receita

Uma receita pode ser apagada da máquina de café se ela está na lista de receitas da máquina. Deve haver uma forma de saber se a receita foi apagada da máquina ou não.

Repor estoque dos ingredientes da máquina de café

Os estoques dos ingredientes das máquinas podem ser repostos. As unidades de cada ingrediente são representadas por números inteiros e positivos. O limite máximo de estoque para cada ingrediente deve ser de no máximo 100 unidades. Os estoques dos ingredientes só são consumidos na máquina quando uma

receita é produzida. A máquina já começa com um estoque de 20 unidades para cada ingrediente. Deve haver uma forma de saber se o acréscimo de unidades de um ingrediente foi realizado com sucesso.

Verificar estoque de ingredientes
Deve ser possível verificar quantas unidades existem de cada ingrediente

Preparar receita
Uma receita deve ser preparada de acordo com a seleção de um usuário e de um valor inserido em centavos para pagar pela bebida. Se a receita existe, e o valor pago é o suficiente, então a bebida deve ser feita, as unidades utilizadas para fazer a bebida devem ser subtraídas do estoque de cada ingrediente, e o troco (se houver) deve ser retornado para o cliente. Se o cliente não inserir dinheiro suficiente, a bebida não deve ser feita e o cliente deve saber que o valor foi insuficiente. Se a quantidade de ingredientes na máquina não for o suficiente para fazer a receita, a receita não deve ser feita e o cliente deve ser avisado do motivo. Sempre que não for possível fazer a bebida, o valor pago pelo cliente deve ser devolvido.

Obter receitas
Deve ser possível obter todas as receitas cadastradas na máquina

3 - Descrição das operações disponíveis

O software para controlar a máquina de café foi implementado em Java e possui as seguintes operações públicas que devem ser usadas pela interface física da máquina de café.

Classe Recipe

`Recipe(String name, int price, int amtCoffee, int amtMilk, int amtSugar, int amtChocolate) throws InvalidValueException`
Entrada: nome, preço, quantidade de café, quantidade de leite, quantidade de açúcar, quantidade de chocolate
Efeito: Criação da receita com os valores de entrada
Exceção: valor inválido de preço, amtCoffee, amtMilk, amtSugar ou amtChocolate

Classe CoffeeMaker

`boolean addRecipe(Recipe r) throws InvalidValueException`
Entrada: Recipe é uma classe que contém os seguintes atributos: name, price, amtCoffee, amtMilk, amtSugar, amtChocolate. A classe Recipe um construtor com parâmetros public Recipe(String name, int price, int amtCoffee, int amtMilk, int amtSugar, int amtChocolate)
Saída: true, se a receita for adicionada com sucesso; false, caso contrário.
Exceção: uma exceção é lançada quando um valor não permitido é lançado para algum item da receita (quantidade de ingredientes ou preço).

`boolean deleteRecipe(String recipeName) throws RecipeException {`
Entrada: nome da receita a ser removida
Saída: true, se a receita for removida com sucesso; false, caso contrário.
Exceção: uma exceção é lançada se a receita a ser deletada não existe.

`void addCoffeeInventory(int amtCoffee) throws InvalidValueException`
Entrada: quantidade do elemento a ser inserido no inventário.
Efeito: a quantidade do elemento em estoque é somada à quantidade recebida por parâmetro.
Exceção: uma exceção é lançada se o valor inserido é incompatível com o permitido pelo inventário

`void addMilkInventory(int amtMilk) throws InvalidValueException`
Entrada: quantidade do elemento a ser inserido no inventário.
Efeito: a quantidade do elemento em estoque é somada à quantidade recebida por parâmetro.
Exceção: uma exceção é lançada se o valor inserido é incompatível com o permitido pelo inventário

`void addSugarInventory(int amtSugar) throws InvalidValueException`
Entrada: quantidade do elemento a ser inserido no inventário.
Efeito: a quantidade do elemento em estoque é somada à quantidade recebida por parâmetro.
Exceção: uma exceção é lançada se o valor inserido é incompatível com o permitido pelo inventário

`void addChocolateInventory(int amtChocolate) throws InvalidValueException`
Entrada: quantidade do elemento a ser inserido no inventário.
Efeito: a quantidade do elemento em estoque é somada à quantidade recebida por parâmetro.
Exceção: uma exceção é lançada se o valor inserido é incompatível com o permitido pelo inventário

`int checkCoffeeInventory()`
Saída: quantidade em estoque do produto consultado

`int checkMilkInventory()`
Saída: quantidade em estoque do produto consultado

`int checkSugarInventory()`
Saída: quantidade em estoque do produto consultado

`int checkChocolateInventory()`
Saída: quantidade em estoque do produto consultado

`int makeCoffee(String recipeName, int amtPaid) throws RecipeException, InsufficientAmountOfMoneyException, InventoryException`
Entrada: nome da receita a ser feita e a quantidade de dinheiro inserida na máquina
Saída: troco.
Exceções: receita inexistente, dinheiro insuficiente ou ingredientes insuficientes

`Vector<Recipe> getRecipes()`
Saída: um vetor com as receitas cadastradas na máquina

Criação dos casos de teste

Elaborem um plano de teste antes de escrever os casos de teste para as operações disponíveis. Isso significa que vocês devem olhar as especificações e definir quais são as classes de equivalência para cada situação e também os valores limites. Em seguida escrevam os casos de teste para testar cada operação.
Dê um nome significativo para cada caso de teste, como, por exemplo, testAddInvalidAmountOfMilk() e testAddValidAmountOfMilk(), para que o próprio teste seja uma documentação do comportamento esperado do sistema.

Detalhes da entrega:

- **Em dupla**
- **Até 29/09**
- **PDF no edisciplinas**