Relatório

**Introdução:**

Em uma ilha tropical pequena, a gasolina dos postos da ilha são um resultado da mistura de 3 componentes, que é feita num único centro de distribuição, encomendado pelos postos.

A partir dos dados recebidos, sobre os percentuais definidos dos componentes e regras que devem ser seguidas, deve-se implementar uma classe DepComb com sua estrutura já feita, com os métodos “recebeAditivo”, “recebeGasolina”, “recebeAlcool” e “encomendaCombustivel” que devem ser implementados pelo grupo.

**Recebendo a carga para cada componente:**

Os respectivos métodos “recebeAditivo”, “recebeGasolina” e “recebeAlcool” servem para receber a carga de cada componente e reservar nos seus respectivos tanques, retornando o quanto foi possível ser armazenado no tanque de acordo com o limite de cada um. Se a quantidade recebida for invalida, retorna -1.

Os métodos “recebeAditivo” e “recebeGasolina” foram implementados da mesma forma, apenas mudando o nome das variáveis. Nesses dois métodos, primeiro é testado se a quantidade recebe é invalido, ou seja, menor que zero, se for, esses métodos retornam -1, caso contrário, é calculado o quanto de espaço ainda está sobrando no tanque, colocando esse valor numa variável chamada maxAdd. Caso a quantidade recebido de combustível ocupar menos ou o mesmo espaço que resta no tanque, é adicionado essa quantidade ao tanque e retornado esse valor, no entanto, se a quantidade recebida exceder o espaço de sobra do tanque, é adicionado o que resta para encher o tanque e retornado maxAdd.

Já para o método “recebeAlcool”, é feito o mesmo teste para saber se a quantidade recebida é invalida, retornando zero caso afirmativo, do contrário, é preciso testar se essa quantidade é ímpar, pois para esse método, é preciso que o combustível seja divido entre dois tanques, por tanto, se for ímpar, é reduzido -1 do combustível. Após esse processo, é colocado na variável maxAdd o calculo de quanto ainda sobra de combustível dos dois tanques de álcool junto, e se esse a quantidade de combustível for menor ou igual ao resultado calculado, dividimos essa quantidade e colocamos o resultado nos dois tanques de álcool, retornando o que foi colocado nos tanques somados, se não for o caso, em vez disso, é colocado a divisão do máximo que os tanques suportam juntos em cada um, e esse máximo é retornado pelo método.

**Encomendando combustível:**

Quando um posto encomenda combustível, é usado o método “encomendaCombustivel”, que recebe quantidade solicitada e se essa encomenda é uma emergência nacional ou não. Se por acaso uma encomenda não puder ser feita, devesse retornar um arranjo com o valor -1 no componente que falta e 0 na posição dos demais, caso a quantidade for um número invalido, retorna -2.

Primeiro é verificado se o valor do combustível solicitado é invalido, se for é retornado um vetor de tamanho 4, contendo o valor -2 na primeira posição e zero nas demais, caso contrário, um vetor de 4 posições é criado para armazenar o quanto de combustível sobrou nos tanques após encaminhar a encomenda. Em seguida, é calculado o quanto é necessário de cada componente para a encomenda, e então, verificado se os tanques contêm o suficiente. Se por acaso a encomenda não for emergencial, é considerado que 25% das capacidades de cada tanque não pode ser usado, portanto, se um dos componentes precisos tiver que usar os 25% que só se usaria numa emergência para ter o suficiente para encomenda, é considerado que ela não pode ser comprida, então retornando um vetor que contem -1 na posição que corresponde ao tanque com quantidade insuficiente e zero nas demais. Se tudo ocorrer bem e os tanques conterem as quantidades solicitadas de cada componente da encomenda, é descontado dos tanques essas quantidades e o método retorna um vetor com o que sobrou em cada tanque após a encomenda ser encaminha.

**Casos de teste:**

Para a criação dos casos de teste foram usados os métodos de limite apresentados abaixo. Com esses métodos cobrimos o máximo possível dos valores limite que exercitam determinadas partes do código:

Recebe Aditivo

Qtdade <= 500

In-points: [0..499]

Out-points: [500..]

Off-point: 499

On-point: 500

|  |  |  |  |
| --- | --- | --- | --- |
| Caso | Classe de Teste | Resultado esperado | Resultado obtido |
| Método recebe 1000  Tanque com 500 | RecebeAditivo | 0 |  |
| Método recebe 300  Tanque com 200 | RecebeAditivo | 300 |  |
| Método recebe -1  Tanque com 500 | RecebeAditivo | -1 |  |
| Método recebe 100  Tanque com 500 | RecebeAditivo | 0 |  |
| Método recebe 1000  Tanque com 400 | RecebeAditivo | 100 |  |

Recebe Alcool

Qtdade <= 2500

In-points: [0..2499]

Out-points: [2500..]

Off-point: 2499

On-point: 2500

|  |  |  |
| --- | --- | --- |
| Caso | Classe de Teste | Resultado esperado |
| Método recebe 1000  Tanque com 2500 | RecebeAlcool | 0 |
| Método recebe 300  Tanque com 2000 | RecebeAlcool | 300 |
| Método recebe -1  Tanque com 2500 | RecebeAlcool | -1 |
| Método recebe 100  Tanque com 2500 | RecebeAlcool | 0 |
| Método recebe 1000  Tanque com 2400 | RecebeAlcool | 100 |
| Método recebe 99  Tanque com 2400 | RecebeAlcool | Coloca 49 em cada tanque e retorna 98 |

Recebe Gasolina

Qtdade <= 10000

In-points: [0..9999]

Out-points: [10000..]

Off-point: 9999

On-point: 10000

|  |  |  |
| --- | --- | --- |
| Caso | Classe de Teste | Resultado esperado |
| Método recebe 1000  Tanque com 10000 | RecebeGasolina | 0 |
| Método recebe 300  Tanque com 9000 | RecebeGasolina | 300 |
| Método recebe -1  Tanque com 1000 | RecebeGasolina | -1 |
| Método recebe 100  Tanque com 10000 | RecebeGasolina | 0 |
| Método recebe 1000  Tanque com 9900 | RecebeGasolina | 100 |