	Tipo de Prova		Ano letivo	Data
	Exame de Época Normal (Prático)		2017-2018	18-01-2018
	Curso		Hora	
	Licenciatura em Engenharia Informática		14:30	
ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Unidade Curricular		Duração	
	Engenharia de Software II		120 minutos	

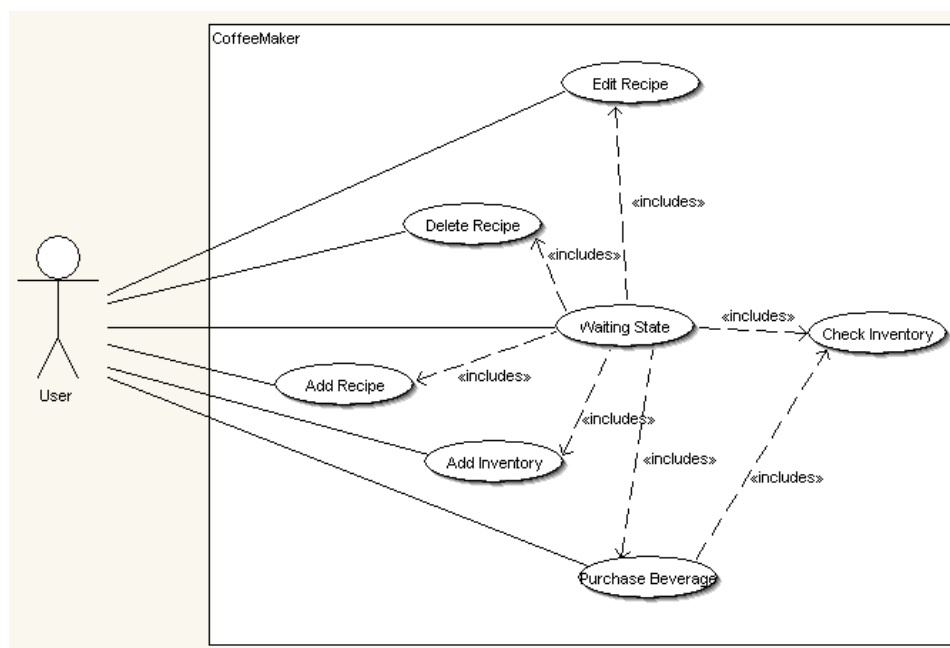
Observações

- O presente exame é **sem consulta!**
- Antes de iniciar a leitura do enunciado, preencha as folhas de exercício que lhe foram fornecidas, com a sua informação pessoal.
- **Coloque o seu número e nome no enunciado que deverá entregar no final do exame.**
- Em cada questão poderá encontrar a respetiva cotação, em valores, entre 0 – 20.
- A implementação dos testes de acordo com os casos de teste especificados deverá ser implementada no computador com recurso ao NetBeans e JUnit Framework. Posteriormente deverá submeter os resultados através do moodle em actividade criada para o efeito.
- **Qualquer questão que não responda, deverá fornecer essa indicação na folha de exercício.**
- Os ficheiros necessários para a realização do presente exame (código fonte + JavaDoc) encontram-se no moodle.
- **As questões de escolha múltipla não serão consideradas se não se encontrarem devidamente justificadas.**
- **Relativamente às questões de escolha múltipla será descontado (1/4)*cotação da pergunta por cada resposta errada.**
- APENAS SE ESCLARECEM DÚVIDAS RELACIONADAS COM DIFICULDADES DE INTERPRETAÇÃO DO ENUNCIADO OU CASO SE VERIFIQUE LAPSO NA ELABORAÇÃO DO MESMO.
- Nota mínima: 9,5 valores


O docente: Cristóvão (cds@estg.ipp.pt)

GRUPO I

Considere uma máquina de venda automática de café, denominada doravante de *CoffeeMaker* (código disponível para download no moodle). A máquina é carregada com um conjunto de ingredientes base e, de acordo com os ingredientes disponíveis é possível fazer uma bebida escolhendo adequadamente as doses de cada ingrediente. Observe o diagrama de use cases referente à máquina *CoffeeMaker*.



De acordo com o Diagrama a máquina permite:

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame de Época Normal (Prático)	Ano letivo 2017-2018	Data 18-01-2018
	Curso Licenciatura em Engenharia Informática	Hora 14:30	
	Unidade Curricular Engenharia de Software II	Duração 120 minutos	


- **Add Recipe** – Apenas 3 receitas podem ser adicionadas na máquina. Uma “receita” consiste em: i) nome; ii) preço; iii) unidades de café; iv) unidades de leite; v) unidades de açúcar; vi) unidades de chocolate;
- O nome da receita deverá ser único na lista de receitas. O preço deverá ser um inteiro. Uma mensagem é apresentada mencionando se a receita foi ou não adicionada com sucesso. Após a receita ter sido adicionada a máquina regressa ao estado de espera (“Waiting State”).
- **Delete Recipe** – Uma receita pode ser eliminada da máquina, mas apenas se ela existir na lista de receitas. As receitas são listadas pelo nome. Após a operação, uma mensagem surgirá a informar do resultado (sucesso/insucesso).
- **Edit Recipe** – Uma receita pode ser editada desde que exista na lista de receitas.
- **Add Inventory** – O Inventário pode ser construído adicionando á máquina, em qualquer altura, ingredientes (items). Os ingredientes que poderão ser adicionados são: i) coffee (café); ii) milk (leite); iii) sugar (açúcar); iv) chocolate;
- Os ingredientes do inventário são medidos em unidades de valores inteiros. A única forma de remover ingredientes do inventário será comprando uma bebida (**Purchase Beverage**). Após a operação, uma mensagem surgirá a informar do resultado (sucesso/insucesso).
- **Check Inventory** – O inventário pode ser verificado a qualquer altura. As unidades de cada ingrediente serão apresentadas. Após a operação, uma mensagem surgirá a informar do resultado (sucesso/insucesso).
- **Purchase Beverage** – O utilizador selecciona uma bebida e insere uma determinada quantia em dinheiro. O valor correspondente á quantia de dinheiro deverá ser um inteiro. Se a bebida estiver no livro de receitas (**Recipe Book**) e se o utilizador inseriu dinheiro suficiente, a bebida será fornecida e o troco é retornado. O dinheiro do utilizador será devolvido na totalidade se não existir produto suficiente em stock. Após a operação, uma mensagem surgirá a informar do resultado (sucesso/insucesso).

Após a leitura cuidadosa da descrição do problema, realize os seguintes exercícios:

1. [cotação: 4 valores]
Após identificar todas as pré-condições do sistema prepare, de acordo com a técnica adequada, o conjunto de casos de teste que cobrem essas pré-condições. Justifique a sua abordagem.
2. [cotação: 4 valores]
Numa abordagem aos testes de software, e de acordo com as funcionalidades identificadas, elabore uma tabela com as classes de equivalência válidas e inválidas.
3. [cotação: 4 valores]
Prepare 5 casos de teste baseados nas tabelas com as classes de equivalência identificadas anteriormente. Não repita casos de teste identificados em 1.
4. [cotação: 4 valores]
Implemente (no NetBeans) os casos de teste identificados anteriormente.

GRUPO II

Observe o pseudo-código que segue:


 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame de Época Normal (Prático)	Ano letivo 2017-2018	Data 18-01-2018
	Curso Licenciatura em Engenharia Informática	Hora 14:30	
	Unidade Curricular Engenharia de Software II	Duração 120 minutos	

Considere o exemplo de uma aplicação para calcular o total de uma fatura de um serviço de telemóvel de acordo com a sua utilização. No exemplo o valor total a ser cobrado obedece a um conjunto de regras. Caso a fatura seja superior a 100€, o cliente terá um desconto de 10%.

```
public static double calculateBill(int Usage)
{
    double Bill = 0;
    if (Usage > 0)
    {
        Bill = 40;
    }
    if (Usage > 100)
    {
        if (Usage <= 200)
        {
            Bill = Bill + (Usage - 100) * 0.5;
        }
        else
        {
            Bill = Bill + 50 + (Usage - 200) * 0.5;
            if (Bill >= 100)
            {
                Bill = Bill * 0.9;
            }
        }
    }
    return Bill;
}
```

Das questões que se seguem, escolha a resposta correta e justifique-a á luz das técnicas/métodos estudados relacionados com testes de software.

- [cotação: 1 valores]
O valor referente à cyclomatic complexity do programa é:
a) 10
b) 9
c) 8
d) Nenhuma das anteriores
- [cotação: 1 valores]
Relativamente ao pseudo-código apresentado na questão anterior, indique o número de casos de teste necessários para garantir 100% de cobertura dos ramos.
a) 5
b) 7
c) 9
d) Nenhuma das anteriores
- [cotação: 1 valores]
Considere um programa que calcula o ponto de intersecção de duas linhas retas e apresenta o resultado. O programa lê dois pares de inteiros (m1, c1) e (m2, c2) que definem duas linhas retas respetivamente a partir da fórmula $y=mx + c$. Das opções seguintes, selecione aquela que corresponde a uma classe de equivalência possível para especificação de casos de teste.:
a) (m1=m2, c1≠c2) 1 1 1
b) (c1 ≠ c2, c2 ≠ m1, m1 ≠ m2)
c) (c1 = c2, m1 = NULL)
d) nenhuma das anteriores 1 1 1

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame de Época Normal (Prático)	Ano letivo 2017-2018	Data 18-01-2018
	Curso Licenciatura em Engenharia Informática	Hora 14:30	
	Unidade Curricular Engenharia de Software II	Duração 120 minutos	

4. [cotação: 1 valores]

Observe o seguinte código em java e os respetivos casos de teste, considerando os elementos (variáveis) de input (age, sex, married):

```
premium = 500;
if ((age<25) && (sex==male) && (!married)) {
    premium += 1500;
} else {
    if (married || (sex==female)) {
        premium -= 200;
    }
    if ((age>45) && (age<65)) {
        premium -= 100;
    }
}
```

Casos de Teste:

```
TesteCase #1: (25, male, false)
TesteCase #2: (50, female, true)
TesteCase #3: (50, male, true)
TesteCase #4: (75, male, true)
```

Os casos de teste apresentados garantem:

- a) 100 % cobertura de ramos [1]
[SEP]
- b) 100% de cobertura de expressões [1]
[SEP]
- c) Nenhuma das anteriores [1]
[SEP]

Boa sorte.