UAlg ISE UNIVERSIDADE DO ALGARVE INSTITUTO SUPERIOR DE ENGENHARIA	**Universidade do Algarve** **Instituto Superior de Engenharia**
	Tecnologias de Mercado ano letivo 2021/2022 Série de exercícios
	Programa Brightstart

1- Implemente o método estático *solution* da classe *StringSplit*, que recebe uma string como parâmetro e a divide retornando um *array* de *strings*, compostas por dois caracteres apenas. Em caso de a string de entrada ter um número ímpar de caracteres é adicionado um "_".

```
StringSplit.solution("Ola");//retorna { "Ol","a_" }
StringSplit.solution("UmaPalavraMesmoMuitoGrande"); // retorna { "Um","aP","al","av","ra","Me","sm","oM","ui","to","Gr","an","de" }
StringSplit.solution("abcde");// retorna { "ab","cd","e_" }
StringSplit.solution("abcdefgh");// retorna { "ab","cd","ef","gh" }
StringSplit.solution("umdoistresquatrocinco");//retorna { "um","do","is","tr","es","qu","at","ro","ci","nc","o_" }
StringSplit.solution("");//retorna {} (Array vazio)
```

2- Escreva um programa para calcular o troco de uma máquina de bebidas. O programa recebe constantemente do *System input*, o valor do produto a comprar e o valor inserido na máquina e escreve no Standard output o troco devolvido pela máquina no menor número de moedas possível:

```
Valor do produto:
1,4
Valor inserido:
2
Troco:  1 moedas de 10 centimos
        1 moedas de 50 centimos
Valor do produto:
2
Valor inserido:
5
Troco:  1 moedas de 1 euros
        1 moedas de 2 euros
Valor do produto:
0,75
Valor inserido:
10
Troco:  1 moedas de 5 centimos
        1 moedas de 20 centimos
        1 moedas de 1 euros
        4 moedas de 2 euros
```

3- Considere o popular jogo *Battleship* (Batalha Naval). O jogo consiste num tabuleiro composto por 100 células dispostas por 10 linhas e 10 colunas, onde na fase de *Setup* os jogadores dispõem cada um dos 5 navios:

Tipo de Navio	Comprimento
Aircraft Carrier	5
Battleship	4
Submarine	3
Cruiser	3
Destroyer	2

Implemente uma aplicação, que reproduza uma partida de Batalha Naval de acordo com as seguintes fases:

Setup:

1. Através do standard input, os jogadores são questionados de onde devem colocar cada um dos Navios. Deve ser fornecido ao standard input, tanto as coordenadas como a orientação em que deve ser colocado o navio. Exemplo:

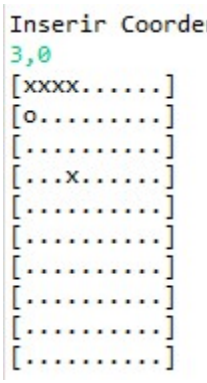
```
Colocar Aircract Carrier
Inserir Orientacao:(H)orizontal/(V)ertical
H
Inserir Coordenadas:
0,0
Colocar Battleship
Inserir Orientacao:(H)orizontal/(V)ertical
V
Inserir Coordenadas:
5,4
```

2. Os Navios devem caber todos no tabuleiro e não se devem em momento algum sobrepor, e caso exista uma posição inválida o utilizador deve ser notificado e deve ser requisitada novamente uma posição correta:

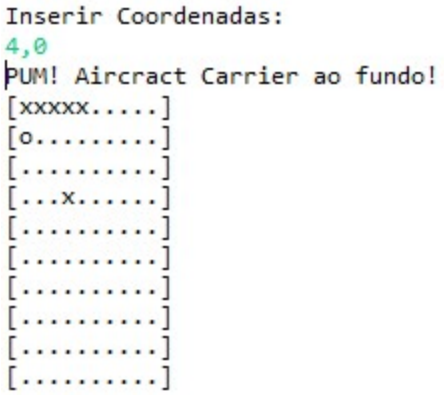
```
Colocar Cruiser
Inserir Orientacao:(H)orizontal/(V)ertical
V
Inserir Coordenadas:
9,9
Nao cabe
Colocar Cruiser
Inserir Orientacao:(H)orizontal/(V)ertical
```

Partida:

1. Na fase de partida à vez, os jogadores devem conseguir consultar o tabuleiro da partida, onde conseguem ver as jogadas que já fizeram e se acertaram ou não num navio:



2. Quando um Navio for afundado, deve ser apresentado no standard output, que navio já não existe mais:



3. O primeiro jogador a afundar todos os navios do adversário vence.

Para este último exercício, o relatório para além da explicação de todas as decisões e clarificações necessárias, deve conter também o Diagrama de classes UML da aplicação.

Bom trabalho.

Data Limite de entrega (relatório e código):

9 de Janeiro de 2022