EXERCÍCIO 1:

Q1-Qual das seguintes proposições é verdadeira quanto ao modificador de acesso protected?

A – Variáveis, métodos e construtores declarados como protected não podem ser acedidos por nenhuma classe;

B – Variáveis, métodos e construtores declarados como protected podem ser acedidos por qualquer classe pertencentes ao mesmo pacote.

C – Variáveis, métodos e construtores declarados como protected na superclasse apenas podem ser acedidos pelas suas subclasses.

D – Nenhuma das proposições anteriores.

Q2 – Qual das seguintes proposições é verdadeira sobre uma superclasse?

A – Variáveis, métodos e construtores declarados como private apenas podem ser acedidos pelos seus objetos.

B – Variáveis, métodos e construtores declarados como protected podem ser acedidos por qualquer uma das suas subclasses.

C – Variáveis, métodos e construtores declarados como public podem ser acedidos por qualquer classe.

D– Qualquer uma das proposições anteriores.

Q3 – O que significa encapsulamento?

A – Encapsulamento é a técnica de definir diferentes métodos ao mesmo tempo .

B – Encapsulamento é a habilidade de um objeto adquirir várias formas.

C – Encapsulamento é a técnica de controlar o tipo de acesso às classes, atributos e métodos, conseguido através dos modificadores de acesso.

D – Nenhuma das proposições anteriores.

Faz uma revisão dos exercícios, da página 9, da ficha nº 1.

EXERCÍCIO 2: Assinala como verdadeiras (V) ou falsas (F) as seguintes proposições (Corrigindo as falsas):

a) Uma subclasse apenas herda os atributos da superclasse. _____.

b) O modificador de acesso protected aplicado a atributos, métodos e construtores numa classe apenas permite que sejam acedidos pelas subclasses dessa classe._____.

c) O polimorfismo aplica-se apenas aos métodos da classe e, por defeito, exige a utilização da herança. _____.

d) No polimorfismo a execução das versões do método não depende da instância da classe que for criada. _____

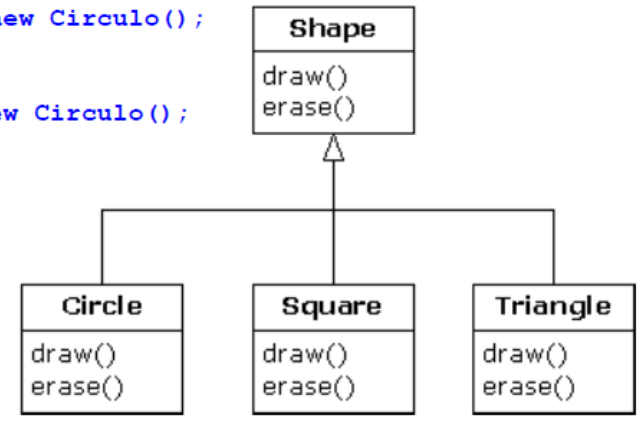
e) Em Java é permitido que uma classe herde as caraterísticas de várias classes. _____

EXERCÍCIO 3: Observa o seguinte diagrama de classes e código:

```
Shape s = new Shape();
s.draw();

Circulo c = new Circulo();
c.draw();

Shape s2 = new Circulo();
s2.draw();
```



3.1 Qual das seguintes caraterísticas da POO se pretende ilustrar? _____

A – Herança
B – Encapsulamento
C – Polimorfismo

3.2 Porquê?

_____
_____
_____

EXERCÍCIO 4:

a) Cria uma classe chamada TV com os seguintes atributos:

```
int canal;
int volume;
boolean ligada;
```

b) Define um construtor que permita atribuir valores aos atributos na criação da instância da classe TV

c) Adiciona os métodos:

- AlterarVolume, com parâmetro do tipo inteiro, que permita aumentar ou diminuir o volume da tv;
- MudarCanal, com parâmetro do tipo inteiro, que permita mudar de canal na tv;
- ligarTV(com parâmetro do tipo booleano, que permita ligar e desligar a tv.

Cabeçalho dos métodos : `public void AlterarVolume (int a)…`

d) Cria uma classe chamada ManusearTV onde se deverá inserir o método *main* para podermos testar a classe TV.

Deverá ser instanciada uma variável do tipo TV para onde os valores iniciais para os atributos sejam:

`canal=1, volume=10, ligada=false e tamanho=21`

e) Cria um método na classe TV, que verifique se a tv está ou não ligada devolvendo o seguinte: "A TV está ligada " ou "A TV está desligada ", caso contrário.

OUTPUT

```
A TV está desligada
```

f) Liga a Tv, muda para o canal 4 e aumenta o volume para 10 usando os métodos criados na classe TV, imprimindo no ecrã a nova situação de forma semelhante à seguinte:

OUTPUT:

```
A TV está ligada  no canal 4 e com o volume 10
```

EXERCÍCIO 5:

Cria um pacote e nome alunos

a) Crie uma classe chamada Aluno, que possua os atributos num, Nome, Turma e contacto.

b) Cria uma classe chamada UsarAlunos onde se utilize um vetor para registar 5 alunos exibindo no ecrã os dados dos alunos inseridos. A impressão de dados no ecrã deverá ser feita através de um método, de nome imprimir, da classe Aluno.

(Consulta a ficha nº 7; Caso não consigas ler os dados das strings, cria uma nova instância da classe scanner : … `=new Scanner(System.in)nextLine();`)

EXERCÍCIO 6:

Refaz o exercício 5, mas com os atributos da classe Aluno, encapsulados. Cria, para isso, um novo pacote de nome Alunos2.

EXERCÍCIO 7: Escreve um programa que dados os nome e cargo (gerente, diretor, operário) calcule o salário do funcionário e o imprima no ecrã do seguinte modo:

Manuel Bichinho o seu cargo é gerente e receberá de salário 900 €

O valor do salário é calculado do seguinte modo:

SB: O salário base é de 550 € e é constante e corresponde ao salário do operário;
Salário do gerente: recebe mais 50% do salário base;
Salário do diretor: recebe mais 70% do salário base.

Caso seja introduzido um cargo diferente dos esperados dever-se-á voltar a pedir a introdução do cargo. Podes usar o método equals que testa o conteúdo dos objetos: `cargo.equals("diretor")`

EXERCÍCIO 8: Escreve um programa que leia 5 valores, os ordene, por ordem decrescente, os imprima no ecrã, divulgando também o maior e o menor deles.

EXERCÍCIO 9:

1. Cria um novo pacote Java de nome **FR, dentro dele uma classe de nome FR** com main, em que NomeApelido é o teu Nome Próprio seguido do teu último Apelido, por exemplo FR_Francisco.

2. Adiciona uma nova classe de nome **Musica** no mesmo pacote.

3. Na classe **Musica** adiciona as seguintes propriedades com visibilidade privada e com o tipo de dados de acordo com o seu possível conteúdo:

a) nome_musica // vai conter o nome da música

b) tempo_seg // vai conter a duração da música em segundos, sem parte decimal

c) nome_cantor

d) tipo_musica // tipo de música "rock", "pop", "clássica", "outra"

4. Na classe **Musica** adiciona um construtor que inicializa todas as propriedades através de 4 parâmetros de entrada.

5. Na classe **Musica** adiciona métodos que altere o valor de cada propriedades (introduzir código de Setter).

6. Na classe **Musica** adiciona métodos que devolvam o conteúdo de cada propriedade (introduzir código de Getter).

7. Adiciona os dados de 2 músicas dos Beatles e uma dos Pink Floyd

8. Na classe **Musica** adicione um método para devolver o conteúdo de todas as propriedades. (introduzir código de toString), de acordo com o seguinte exemplo:

```
Musica: Flying, 480, segundos, cantor: Beatles, Tipo de música:rock
```

Bibliografia:
https://www.w3schools.com/java
https://www.tutorialspoint.com/java/
Jesus, C. (2013). Curso Prático de Java. Lisboa: FCA
Coelho, P (2016). Programação em JAVA – Curso Completo. Lisboa: FCA