UNIVERSIDADE FEDERAL DE UBERLÂNDIA

CURSO REFORÇO POO1

LUCAS FERNANDO
KELLY CRISTINA

MODIFICADORES DE VISIBILIDADE

MODIFICADORES

Indicam o nível de acesso aos componentes internos de uma classe.

01 ?

02 ?

03 ?

QUAIS SÃO OS MODIFICARES?

01 + public

02 ?

03 ?

QUAIS SÃO OS MODIFICARES?

QUAIS SÃO OS MODIFICARES?

01 + public

02 - private

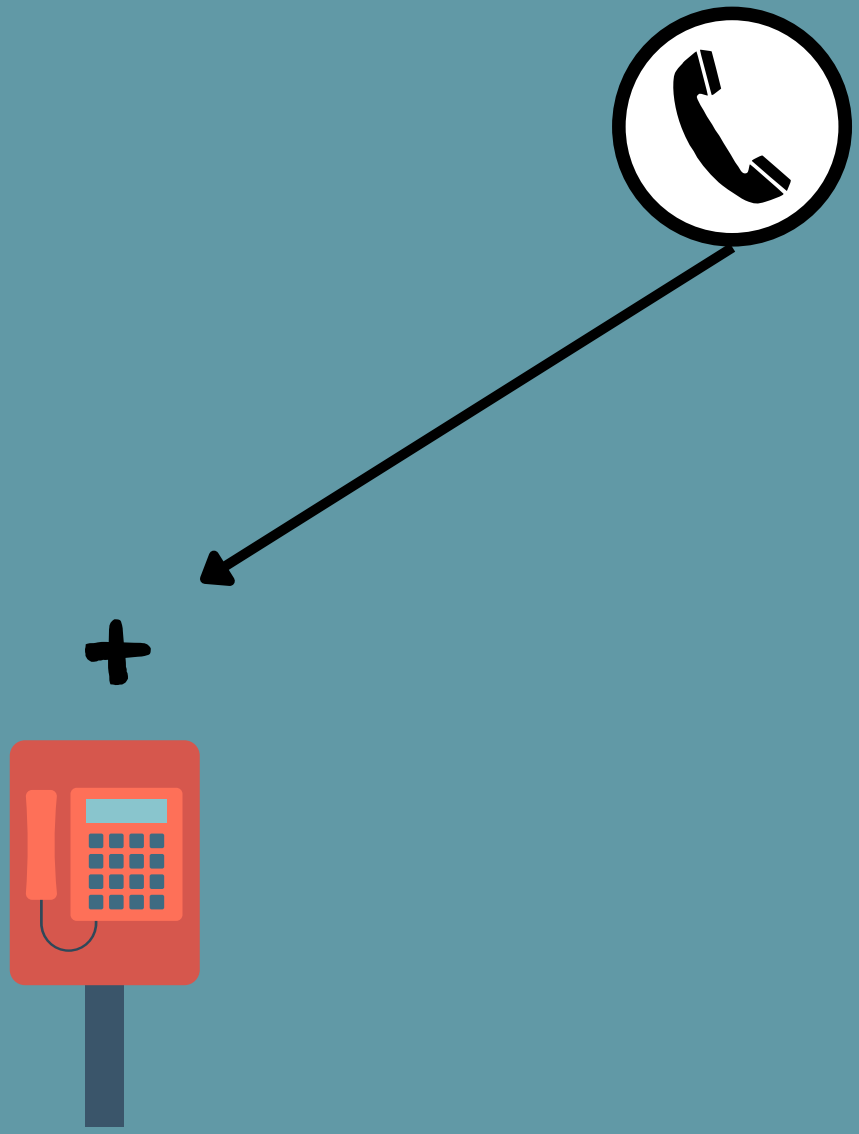
03 ?

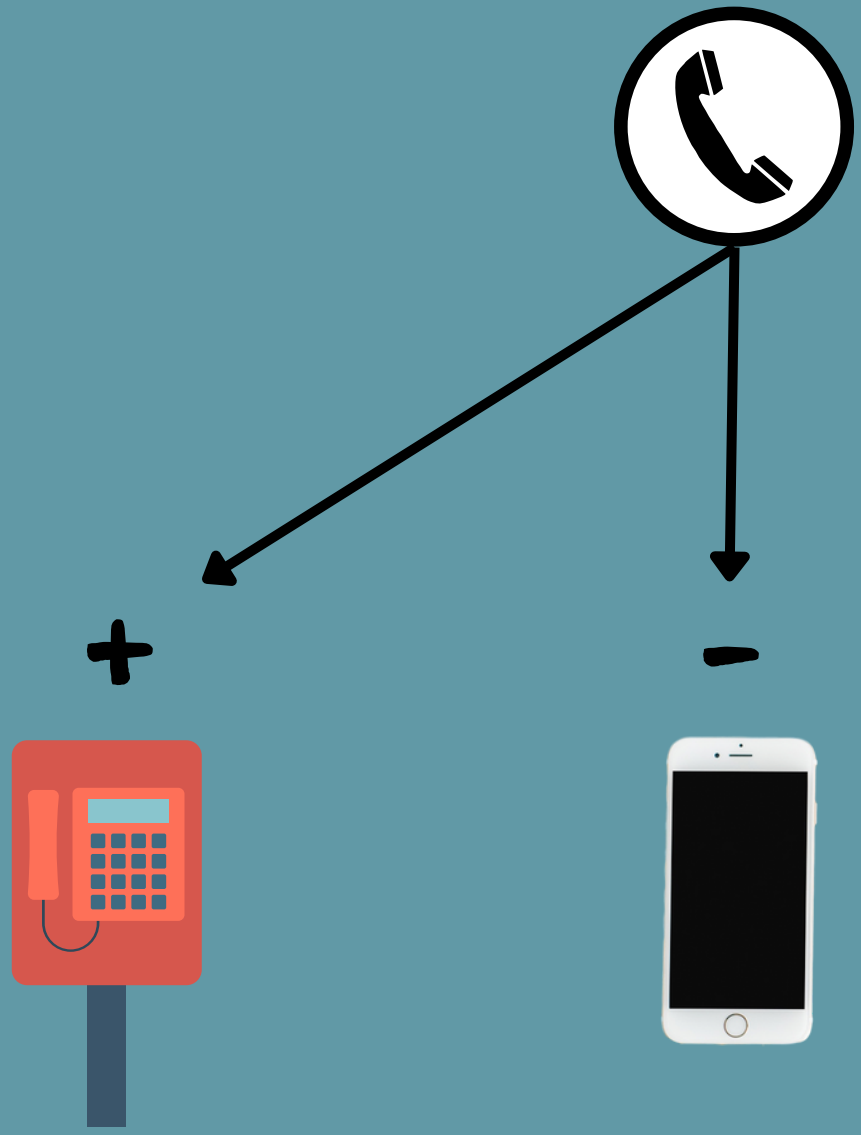
QUAIS SÃO OS MODIFICARES?

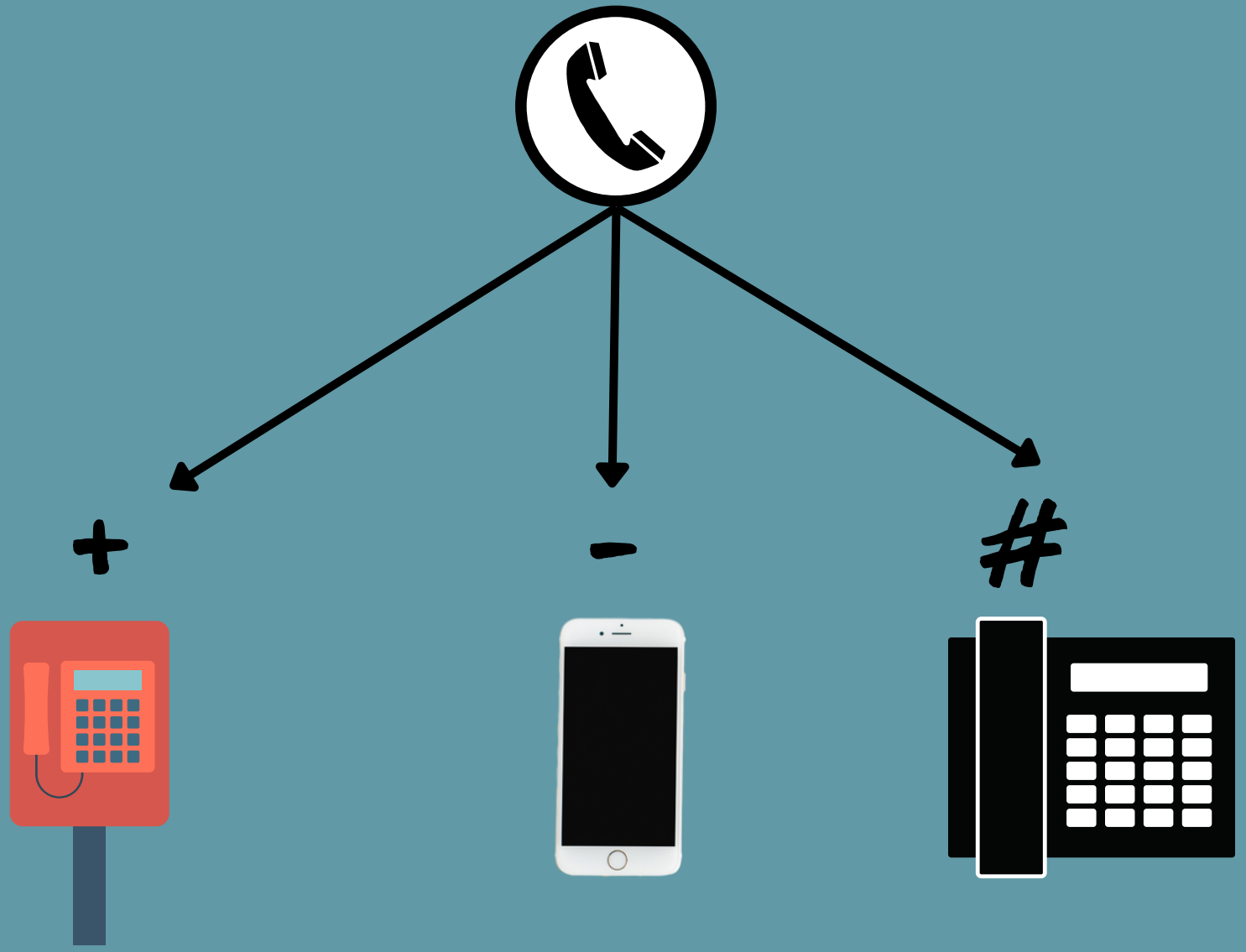
01 + public

02 - private

03 # protected







DEFINIÇÃO EM POO

01 + public

A classe atual e todas as outras classes pode ter acesso a atributos ou métodos.

02 - private

Somente a classe atual pode ter acesso a atributos ou métodos.

03 # protected

A classe atual e todas as outras sub-classes tem acesso aos métodos e atributos.

EXEMPLO


Caneta
+ modelo
+ cor
- ponta
carga
tampada
+ escrever()
+ rabiscar()
+ pintar()
- tampar()
- destampar()

EXEMPLO

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.cor="azul";
    caneta1.modelo="Bic";

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

1

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.cor="azul";
    caneta1.modelo="Bic";

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

1

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.cor="azul";
    caneta1.modelo="Bic";

}
```

```
public String modelo;
public String cor;
protected Integer carga;
protected Boolean tampa;

public void rabiscar(){
    System.out.println("Rabiscando na folha de caderno");
}
private void tampar(){
    System.out.println("Tampando caneta");
}

protected void limpar(){
    System.out.println("Tampando caneta");
}
}
```

1

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.ponta= (float)0.7;

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

2

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.ponta= (float)0.7;

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

2

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.ponta= (float)0.7;

}
```

```
public class Caneta {
    public String modelo;
    private float ponta;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

2

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.carga=50;

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

3

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.carga=50;

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

3

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.carga=50;

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

3

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.rabiscar();

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

4

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.rabiscar();

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

4

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.rabiscar();

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        o na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

4

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.tampar();

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

5

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.tampar();

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

5

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.tampar();

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

5

ESTA CORRETO OU FUNCIONA?

```java
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.limpar();

}
```

```java
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

6

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args)

    Caneta caneta1 = new Caneta();
    caneta1.limpar();

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Tampando caneta");
    }
}
```

6

ESTA CORRETO OU FUNCIONA?

```
public static void main(String[] args)

    Caneta canetal = new Caneta();
    canetal.limpar();

}
```

```
public class Caneta {
    public String modelo;
    public String cor;
    private float ponta;
    protected Integer carga;
    protected Boolean tampa;

    public void rabiscar(){
        System.out.println("Rabiscando na folha de caderno");
    }
    private void tampar(){
        System.out.println("Tampando caneta");
    }

    protected void limpar(){
        System.out.println("Limpando caneta");
    }
}
```

6

GETTER E SETTER

GETTER E SETTER

ESTES MÉTODOS SÃO SELETORES E MODIFICADORES DOS ATRIBUTOS DE SUA CLASSE. ATRAVÉS DELES APLICA-SE UMA REGRA DA ORIENTAÇÃO A OBJETO CHAMADA ENCAPSULAMENTO QUE CONSISTE QUE OS ATRIBUTOS DE UMA CLASSE NÃO PODEM SER ACESSADOS DIRETAMENTE.

GETTER E SETTER

- GETTERS E SETTERS SÃO USADOS PARA PROTEGER SEUS DADOS, ESPECIALMENTE NA CRIAÇÃO DE CLASSES.
- PARA CADA INSTÂNCIA DE VARIÁVEL, UM MÉTODO GETTER RETORNA SEU VALOR, ENQUANTO UM MÉTODO SETTER O DEFINE OU ATUALIZA.

GETTER E SETTER

- GETTERS E SETTERS TAMBÉM SÃO CONHECIDOS COMO MÉTODOS DE ACESSO E DE MODIFICAÇÃO, RESPECTIVAMENTE.

GETTER E SETTER

- POR CONVENÇÃO, GETTERS COMEÇAM COM A PALAVRA "GET" E SETTERS COM A PALAVRA "SET", SEGUIDOS DE UM NOME DE VARIÁVEL.
- EM AMBOS OS CASOS, A PRIMEIRA LETRA DO NOME DA VARIÁVEL SERÁ MAIÚSCULA;

EXEMPLO

```java
public class Caneta {

    public String cor;

    public String getCor() {
        return cor;
    }

    public void setCor(String cor) {
        this.cor = cor;
    }
}
```

```java
public class Caneta {

    public String cor;

    public String getCor() {
        return cor;
    }

    public void setCor(String cor) {
        this.cor = cor;
    }
}
```

GETTER

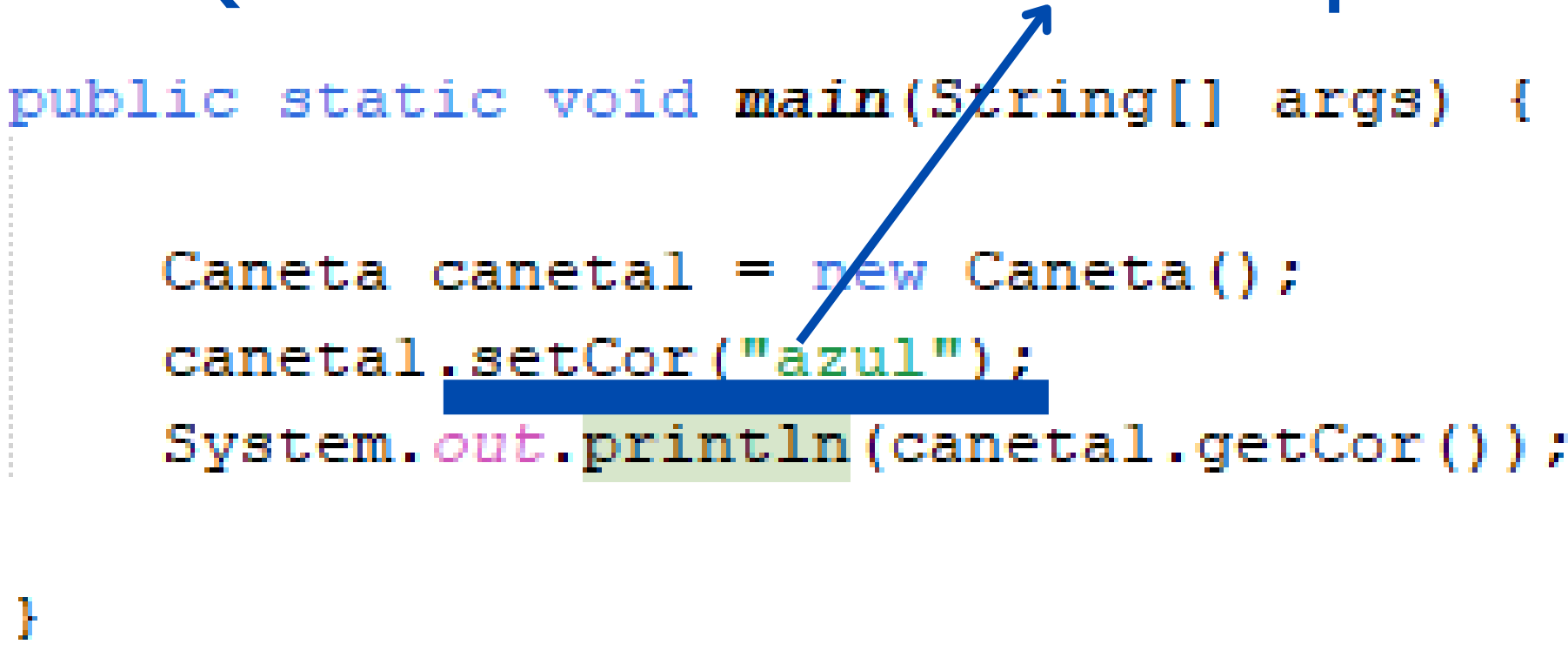
SETTER

```
public static void main(String[] args) {

    Caneta canetal = new Caneta();
    canetal.setCor("azul");
    System.out.println(canetal.getCor());

}
```

Quero modificar a cor da caneta pra azul



```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.setCor("azul");
    System.out.println(caneta1.getCor());

}
```

```
public static void main(String[] args) {

    Caneta canetal = new Caneta();
    canetal.setCor("azul");
    System.out.println(canetal.getCor());

}
```

Quero ver qual cor está nesse atributo

Quero modificar a cor da caneta pra azul

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.setCor("azul");
    System.out.println(caneta1.getCor());

}
```

Quero ver qual cor está nesse atributo

```
public static void main(String[] args) {

    Caneta canetal = new Caneta();
    canetal.setCor("azul");
    System.out.println(canetal.getCor());

}
```

```
run:
azul
```

Saída da operação de setter e getter;

QUE COR ESSE GETTER RETORNA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.setCor("azul");
    caneta1.setCor("vermelha");
    caneta1.setCor("preta");
    caneta1.setCor("rosa");
    System.out.println(caneta1.getCor());

}
```

QUE COR ESSE GETTER RETORNA?

```
public static void main(String[] args) {

    Caneta caneta1 = new Caneta();
    caneta1.setCor("azul");
    caneta1.setCor("vermelha");
    caneta1.setCor("preta");
    caneta1.setCor("rosa");
    System.out.println(caneta1.getCor());

}
```

```
run:
rosa
BUILD SUCCESSFUL (total time: 0 seconds)
```

DICA!

- APENAS TOMEM CUIDADO PARA NÃO SAIREM CIRANDO GET/SET PRA TUDO, POIS TEM ATRIBUTOS QUE NÃO PRECISAM/DEVEM SER ALTERADOS.

RETORNA MESMO VALOR?

```
Caneta canetal = new Caneta();
System.out.println(canetal.getCor());
System.out.println(canetal.getModelo());
System.out.println(canetal.getPonta());
```

```
Caneta canetal = new Caneta();
System.out.println(canetal.cor);
System.out.println(canetal.modelo);
System.out.println(canetal.ponta);
```

```java
Caneta canetal = new Caneta();
System.out.println(canetal.getCor());
System.out.println(canetal.getModelo());
System.out.println(canetal.getPonta());
```

```java
Caneta canetal = new Caneta();
System.out.println(canetal.cor);
System.out.println(canetal.modelo);
System.out.println(canetal.ponta);
```

atributos encapsulados

```
Caneta caneta1 = new Caneta();
System.out.println(caneta1.getCor());
System.out.println(caneta1.getModelo());
System.out.println(caneta1.getPonta());
```

```
Caneta caneta1 = new Caneta();
System.out.println(caneta1.cor);
System.out.println(caneta1.modelo);
System.out.println(caneta1.ponta);
```

acesso direto aos atributos

CONSTRUTOR

CONSTRUCTOR

- Quando usamos a palavra 'new' para criar um objeto de uma determinada classe estamos alocando um espaço na memória.

- Ao fazer isso, o Java requer que algumas variáveis sejam iniciadas com algum valor.

- Esse ato de inicializar, ou construir, é feito pelos construtores;

O QUE ESSE CÓDIGO IMPRIME?

```
public class Caneta {

    public String cor;
    public String modelo;
    public int ponta;
```

```
Caneta caneta1 = new Caneta();
System.out.println(caneta1.getCor());
System.out.println(caneta1.getModelo());
System.out.println(caneta1.getPonta());
```

CONSTRUCTOR

```
Caneta canetal = new Caneta();
System.out.println(canetal.getCor());
System.out.println(canetal.getModelo());
System.out.println(canetal.getPonta());
```

```
run:
null
null
0
BUILD SUCCESSFUL (total time: 2 seconds)
```

- **Ou seja, por padrão, as strings são iniciadas com valores 'null' (nulos) e valores zeros para números.**

CONSTRUCTOR

- **Então construtores são basicamente um método em java.**
- **O construtor padrão não recebe nenhum parâmetro.**
- **Obervação: Podemos criar vários construtores com parâmetros diferentes.**

EXEMPLO

EXEMPLO

```
public class Caneta {

    public Caneta(){

    }
    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

EXEMPLO

```
public class Caneta {

    public Caneta(){

    }
    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

EXEMPLO

Construtor 1

```
public class Caneta {

    public Caneta(){


    }
    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

EXEMPLO

```
public class Caneta {

    public Caneta(){

    }
    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

EXEMPLO

Construtor 1

```
public class Caneta {

    public Caneta(){


    }

    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

Constutor 2

```java
public class Caneta {

    public Caneta(){

    }
    public Caneta(String c , String m){
        this.cor= c;
        this.modelo=m;
    }
```

- **Podemos criar um Construtor sem definir nenhum dado, ou podemos criar uma caneta definindo sua cor e seu modelo por exemplo.**

O QUE ESSE CÓDIGO IMPRIME?


```
Caneta caneta1 = new Caneta();
System.out.println(caneta1.getCor());
System.out.println(caneta1.getModelo());
System.out.println(caneta1.getPonta());
```


```
Caneta caneta2 = new Caneta("azul","bic",7);
System.out.println(caneta2.getCor());
System.out.println(caneta2.getModelo());
System.out.println(caneta2.getPonta());
```

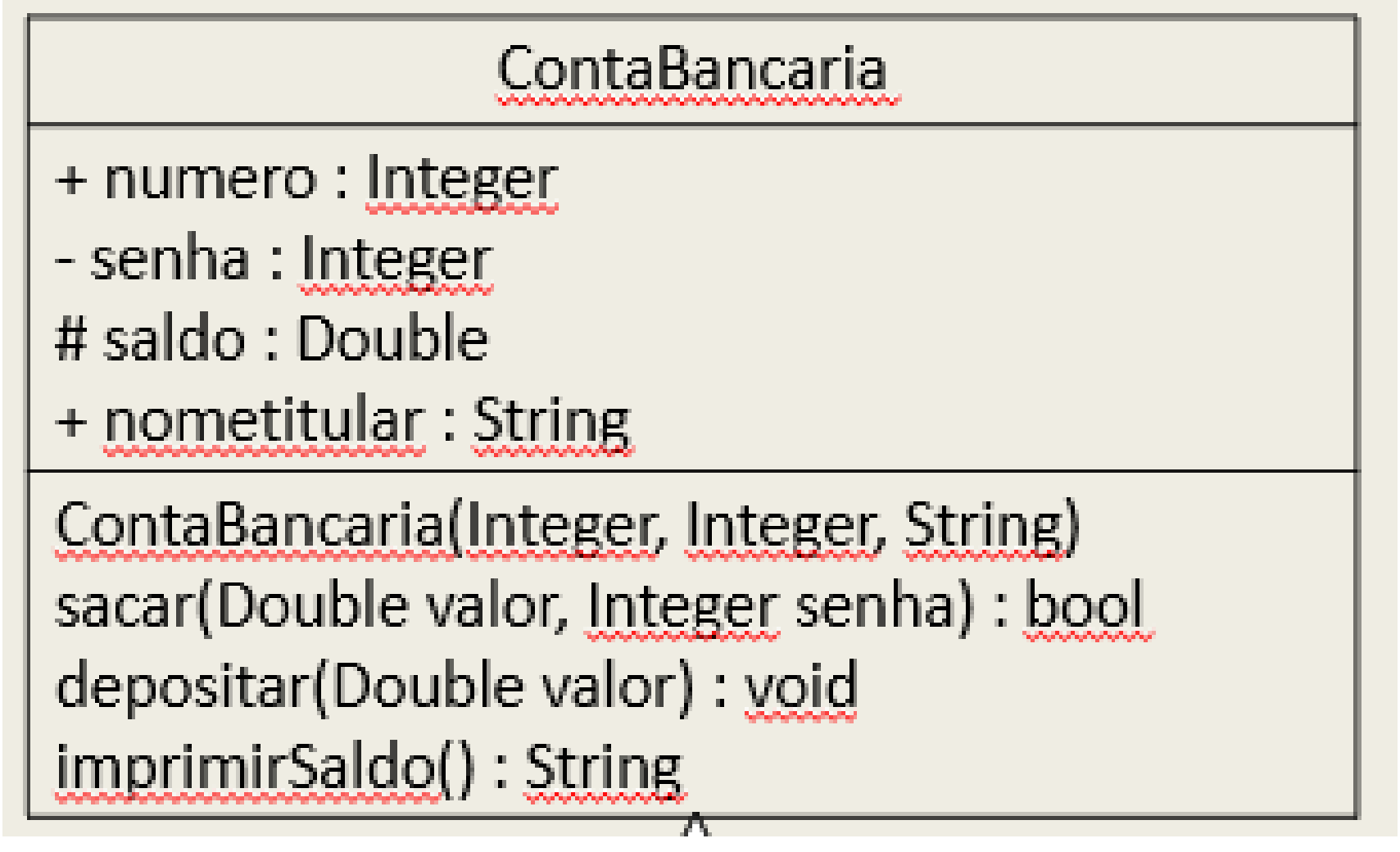
O QUE ESSE CÓDIGO IMPRIME?


```
null
null
0
```


```
azul
bic
7
```

PRÁTICA

PRÁTICA


ContaBancaria
+ numero : Integer
- senha : Integer
saldo : Double
+ nometitular : String
ContaBancaria(Integer, Integer, String)
sacar(Double valor, Integer senha) : bool
depositar(Double valor) : void
imprimirSaldo() : String

REFERÊNCIAS

- GUANABARA (CURSO EM VÍDEO)

- DEV MEDIA

FIM

DÚVIDAS OU PERGUNTAS?