

Nem sempre a importação do bytecode funciona de primeira nas IDEs. Por isso, caso prefira, você também pode fazer o teste do encapsulamento sem uso do Eclipse, IntelliJ ou outra aplicação do gênero.

OBS.: Para título de exemplificação, foi criado um bytecode **Pessoa.class**, que é diferente ao da atividade proposta.

Considere a classe a seguir:

```
public class Pessoa{

    _____String nome;
    _____String sobrenome;
    _____double salario;
    _____String cpf;



    _____void cadPessoa(String nome, String sobrenome, double salario, String cpf)
    {
        this.nome = nome;
        this.sobrenome = sobrenome;
        this.salario = salario;
        this.cpf = cpf;
    }

    _____void impPessoa() {
        System.out.println("\nNome.....: "+ nome);
        System.out.println("\nSobrenome.....: "+ sobrenome);
        System.out.println("\nSalario.....: "+ salario);
        System.out.println("\nCPF.....: "+ cpf);
    }

    _____void calcSalario() {
        salario = salario * 1.50;
    }
}
```

Como saber quais os modificadores de cada atributo e método sem acesso ao código fonte e sem uma IDE?

- 1) Copie o arquivo bytecode fornecido para um diretório qualquer do seu computador.
- 2) Em seguida, crie, no mesmo diretório onde está o bytecode, uma classe de teste utilizando o editor de texto de sua preferência.

Nome	Data de modificação	Tipo	Tamanho
 Pessoa.class	23/03/2022 19:59	Arquivo CLASS	2 KB
 Teste.java	23/03/2022 20:04	Arquivo Fonte Java	1 KB

```
1 public class Teste {  
2  
3     public static void main(String[] args) {  
4  
5  
6     }  
7  
8 }  
9
```

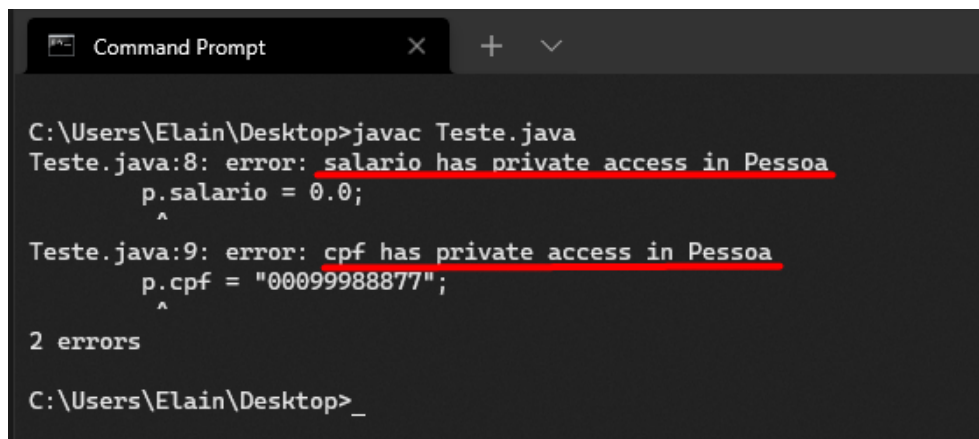
- 3) Feito isso, é hora de começar a criar os objetos e testar os modificadores. Pela documentação fornecida acima, sabe-se que a classe Pessoa possui os atributos: *nome*, *sobrenome*, *salario* e *cpf*. Para testar o acesso a cada um deles, crie o objeto do tipo Pessoa na classe Teste e atribua um valor a cada um deles.

```
1 public class Teste {  
2  
3     public static void main(String[] args) {  
4         Pessoa p = new Pessoa();  
5  
6         p.nome = "teste";  
7         p.sobrenome = "sobrenome";  
8  
9         p.salario = 0.0;  
10        p.cpf = "00099988877";  
11  
12    }  
13  
14 }  
15
```

- 4) Feito isso, salve a classe Teste.java e a compile utilizando a linha de comando.

```
javac Teste.java
```

- 5) Ao fazer isso, note que o terminal mostra uma mensagem de erro.



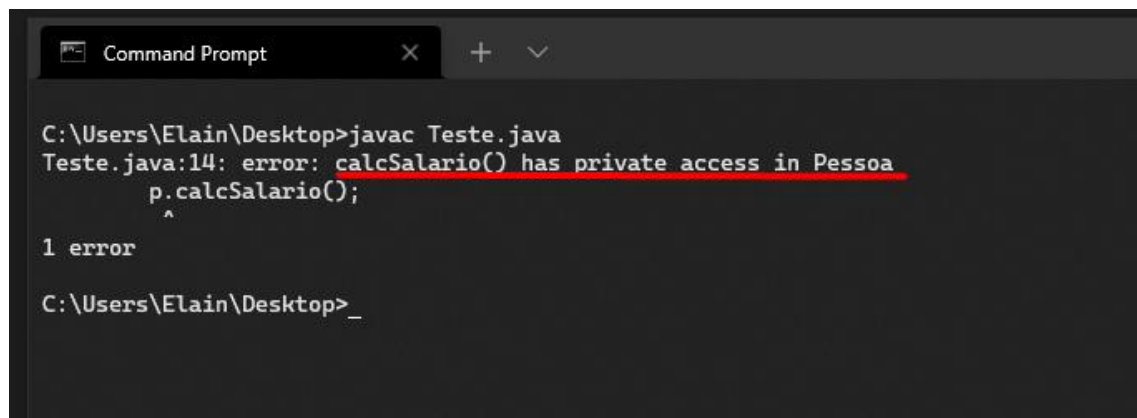
```
Command Prompt  
C:\Users\Elain\Desktop>javac Teste.java  
Teste.java:8: error: salario has private access in Pessoa  
    p.salario = 0.0;  
      ^  
Teste.java:9: error: cpf has private access in Pessoa  
    p.cpf = "00099988877";  
      ^  
2 errors  
C:\Users\Elain\Desktop>_
```

Ou seja, os atributos *salario* e *cpf* possuem modificador de acesso **PRIVATE** na classe *Pessoa*, não sendo possível o acesso direto a eles.

6) O mesmo vale para os métodos.

```
1  public class Teste {  
2  
3      public static void main(String[] args) {  
4          Pessoa p = new Pessoa();  
5  
6          p.cadPessoa("Teste","Atividade1 JavaI",100.00,"22233344400");  
7          p.impPessoa();  
8          p.calcSalario();  
9      }  
10 }
```

7) Ao compilar o código acima...



```
Command Prompt  
C:\Users\Elain\Desktop>javac Teste.java  
Teste.java:14: error: calcSalario() has private access in Pessoa  
    p.calcSalario();  
      ^  
1 error  
C:\Users\Elain\Desktop>_
```

Dessa forma, munido da documentação da classe, é possível saber quais os modificadores de uma classe sem a utilização de uma IDE.