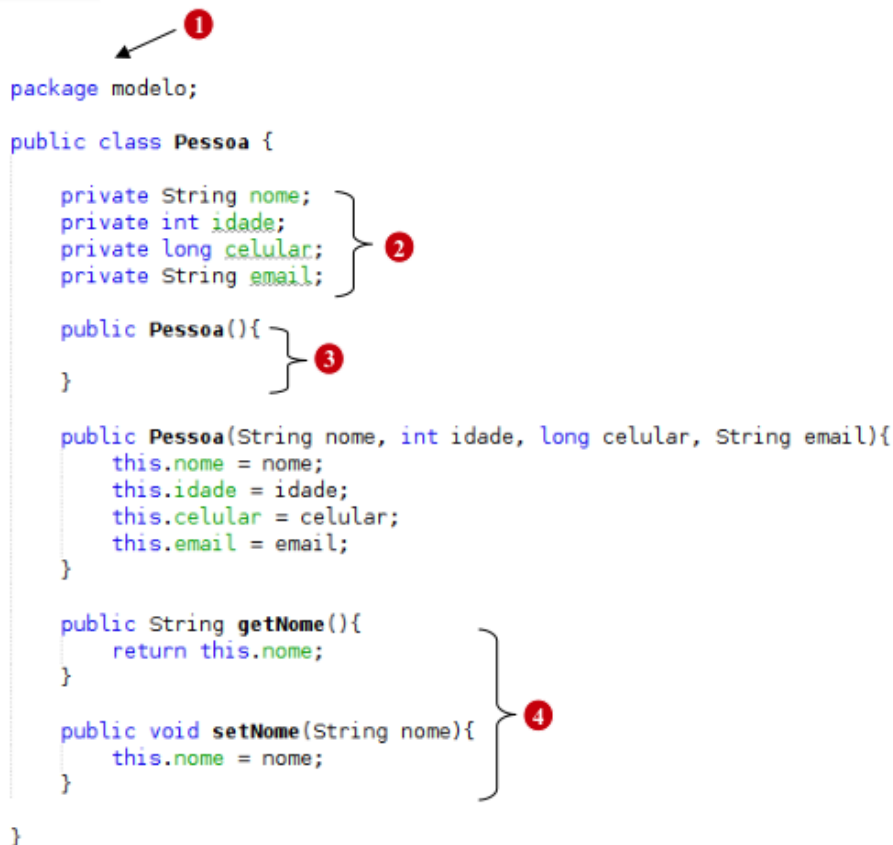


Disciplina: Programação Orientada a Objetos
Prof. Rodrigo R. Silva (rodrigorosa@ifsul.edu.br)
<https://github.com/Prof-Rodrigo-Silva>

Exercícios POO - Lista 4

1. Dada a estrutura da classe abaixo, responda o que significa cada trecho assinalado.



```
package modelo;

public class Pessoa {

    private String nome;
    private int idade;
    private long celular;
    private String email;

    public Pessoa(){

    }

    public Pessoa(String nome, int idade, long celular, String email){
        this.nome = nome;
        this.idade = idade;
        this.celular = celular;
        this.email = email;
    }

    public String getNome(){
        return this.nome;
    }

    public void setNome(String nome){
        this.nome = nome;
    }

}
```

Diagram illustrating the structure of the `Pessoa` class with annotations:

- 1: Points to the `package modelo;` declaration.
- 2: Points to the private attributes (`nome`, `idade`, `celular`, `email`).
- 3: Points to the no-argument constructor (`public Pessoa(){}`).
- 4: Points to the `getNome()` and `setNome()` methods.

2. Responda as seguintes questões com base na classe apresentada na questão anterior.

- Se no código apresentado não tivesse a instrução `package modelo;`, onde deveria estar esta classe na estrutura do projeto?
- Se no lugar de `private` estivesse o modificador `protected`, o que modificaria no código?
- O que acontece se colocarmos `abstract` na declaração da classe?

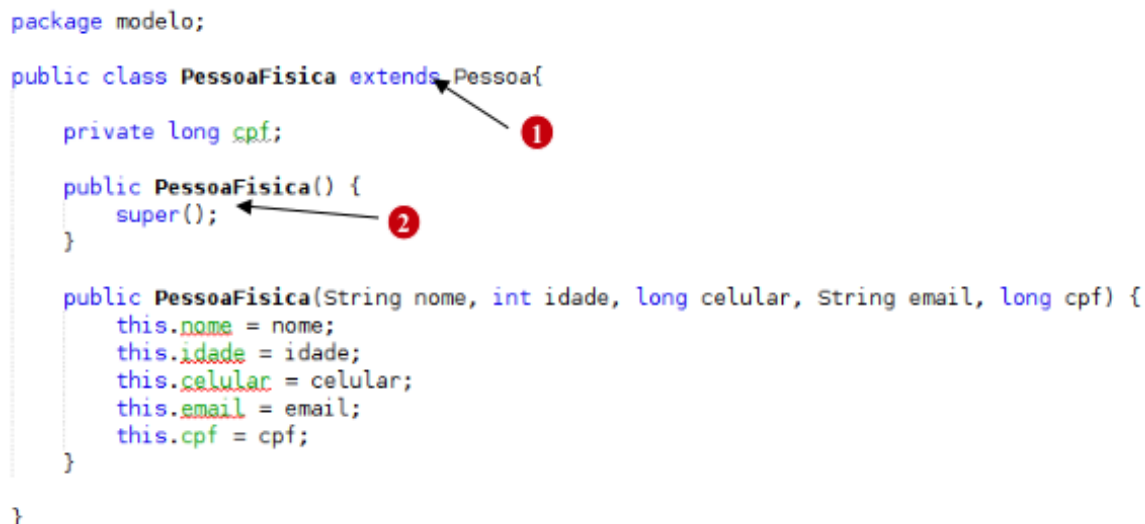
- d) O que é sobrecarga de método? Onde é aplicado este conceito? Qual sua utilidade?
- e) Qual é a finalidade da palavra reservada `this`, empregada em algumas instruções na classe `Pessoa`?
3. Dada a estrutura da classe abaixo, responda o que significa cada trecho assinalado.

```
package modelo;

public class PessoaFisica extends Pessoa{
    private long cpf;

    public PessoaFisica() {
        super();
    }

    public PessoaFisica(String nome, int idade, long celular, String email, long cpf) {
        this.nome = nome;
        this.idade = idade;
        this.celular = celular;
        this.email = email;
        this.cpf = cpf;
    }
}
```



4. Responda as seguintes questões com base no código da classe `PessoaFisica`.
- a) Por que o construtor de `PessoaFisica` está acusando alguns erros?
- b) Quais seriam as soluções para resolver o problema?
- c) O que acontece se colocarmos `final` na declaração da classe?
- d) Além dessa, quais são as outras finalidades da palavra `final`?

```
package modelo;

public abstract class Matematica {

    private static final double PI = 3.14159265359;

    public static double calcularMediaAritmetica(double n1, double n2){
        return (n1 + n2)/2;
    }

    public static double calcularMediaPonderada(double n1, int p1, double n2, int p2){
        return (n1 * p1 + n2 * p2) / (p1 + p2);
    }

    public static double calcularAreaCirculo(double raio){
        return PI * raio * raio;
    }
}
```

5. Com base na classe abaixo, responda as questões.

- a) O que acontece se incluirmos final além de abstract na classe Matematica?
 - b) Que tipo de métodos a classe Matematica contém?
 - c) Além dos tipos de métodos encontrados na classe Matematica, quais são os outros tipos?
 - d) Como chamamos os métodos da classe Matematica em aplicações Java?
6. Dadas as classes Funcionario e Setor, apresentadas abaixo, responda as seguintes questões.

```
package modelo;

public class Funcionario extends PessoaFisica{

    private double salario;
    private Setor setor;

    public Funcionario() {
        super();
    }

    public double getSalario() {
        return salario;
    }

    public void setSalario(double salario) {
        this.salario = salario;
    }

    public Setor getSetor() {
        return setor;
    }

    public void setSetor(Setor setor) {
        this.setor = setor;
    }
}
```

```
package modelo;

public class Setor {

    private String nome;

    public Setor() {

    }

    public Setor(String nome) {
        this.nome = nome;
    }

    public String getNome() {
        return nome;
    }

    public void setNome(String nome) {
        this.nome = nome;
    }
}
```

- a) Qual é a relação entre as duas classe do modelo?
 - b) Como seria o código do construtor sobrecarregado das duas classes?
 - c) Qual linha de código apresenta o nome do setor de um funcionário em um aplicativo Java?
7. O que são interfaces Java? Qual sua utilidade? Em que situações podemos empregar?
8. O que é a classe Object? Escreva um trecho em Java que instancia um objeto do tipo PessoaFisica e, após, referencie como um Object.
9. Qual é a finalidade da coerção de objetos? Exemplifique.
10. Escreva o comportamento dos métodos toString() e equals() na classe Pessoa.