

A thick black L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

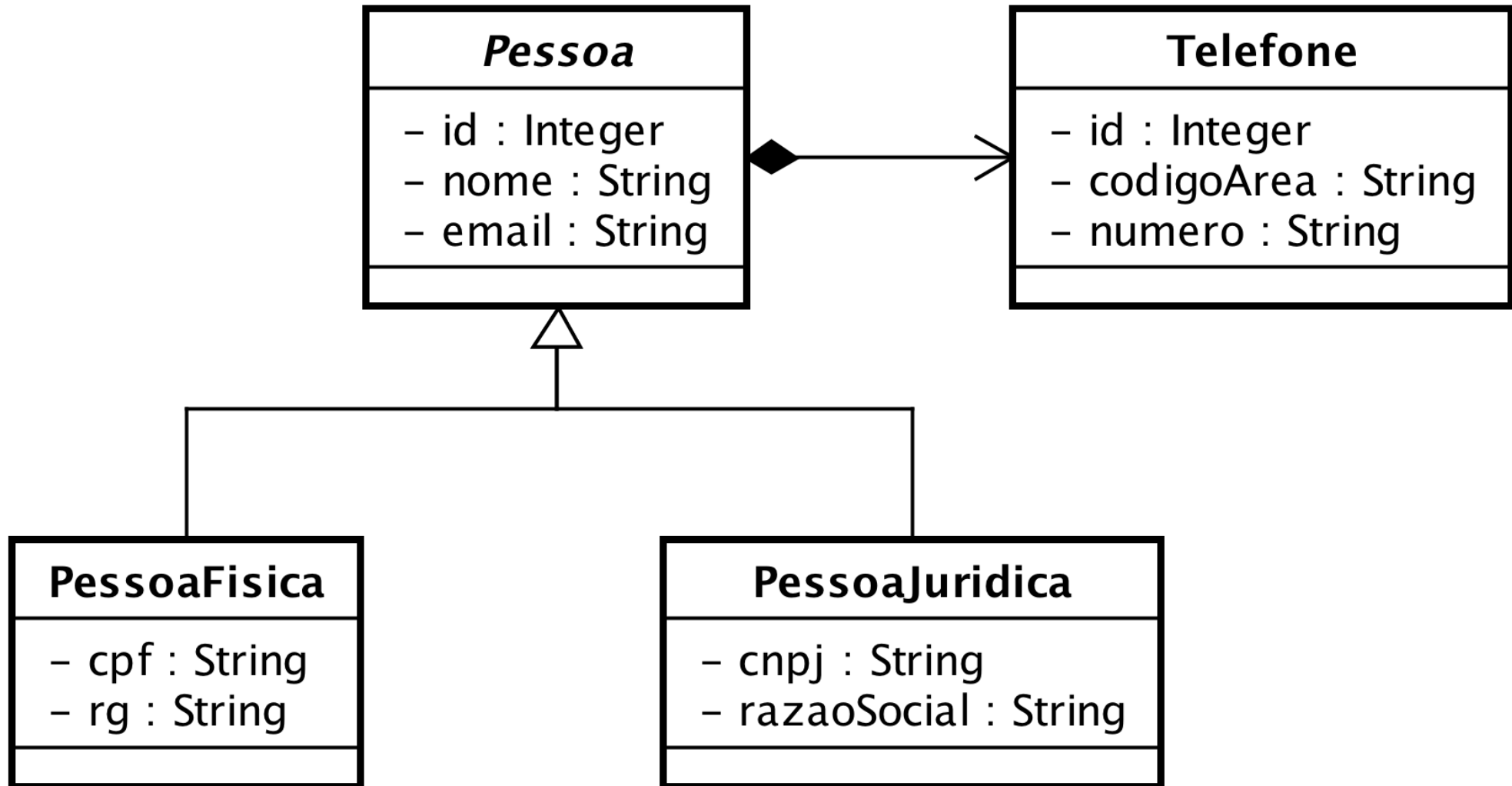
ALGORITMOS E PROGRAMAÇÃO II

Professor: Jânio Elias Teixeira Júnior

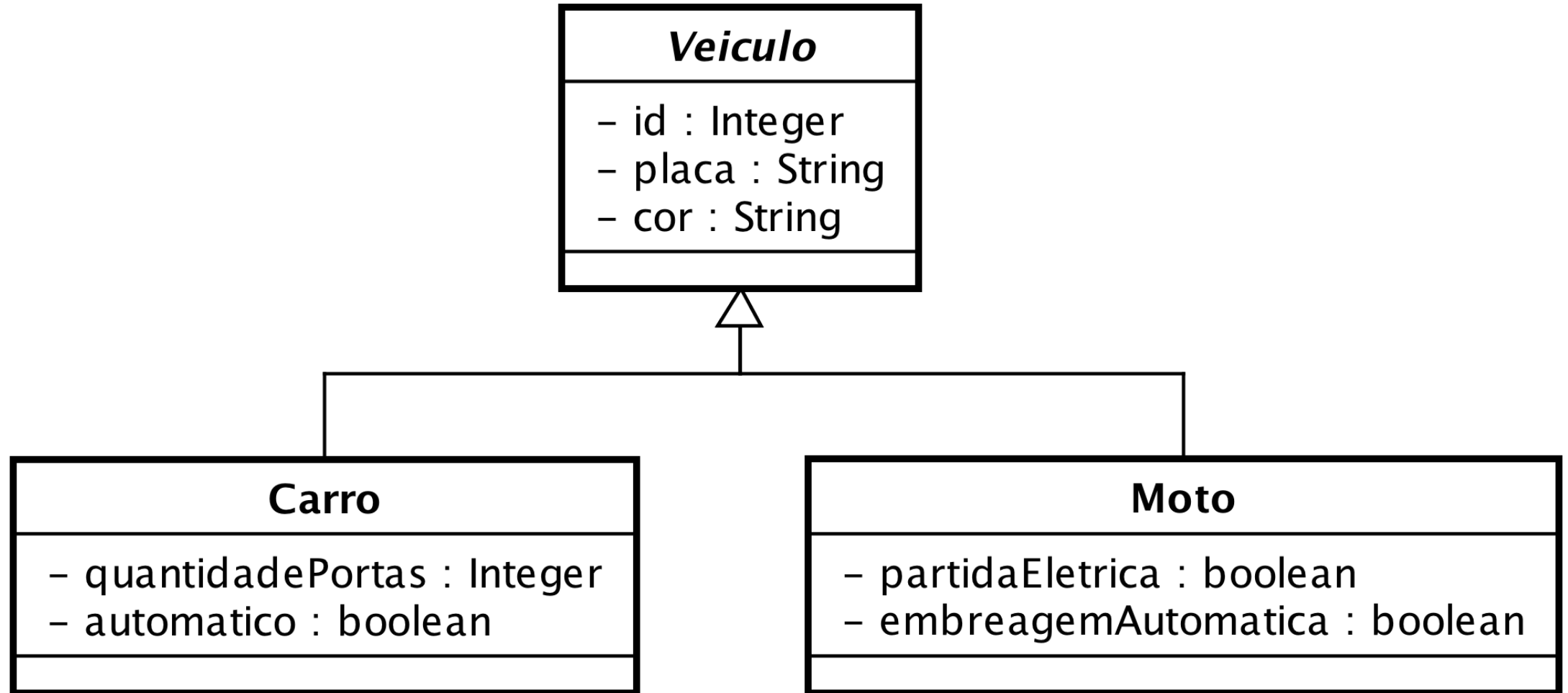


CLASSE ABSTRATA

Classe Abstrata – Exemplo em UML



Classe Abstrata – Exemplo em UML



Classe Abstrata - Conceito

- Uma classe abstrata “não pode ser instanciada”;
 - *Exceto para uma instância anônima.*
- Serve como modelo para uma classe concreta;
 - *São utilizadas como herança.*
- Em Java é utilizado o modificador de acesso ***abstract*** para definir uma classe abstrata.

Classe Abstrata – Exemplo Java

```
public abstract class Veiculo {  
    private Integer id;  
    private String placa;  
    private String cor;  
}
```



MÉTODOS ABSTRATOS



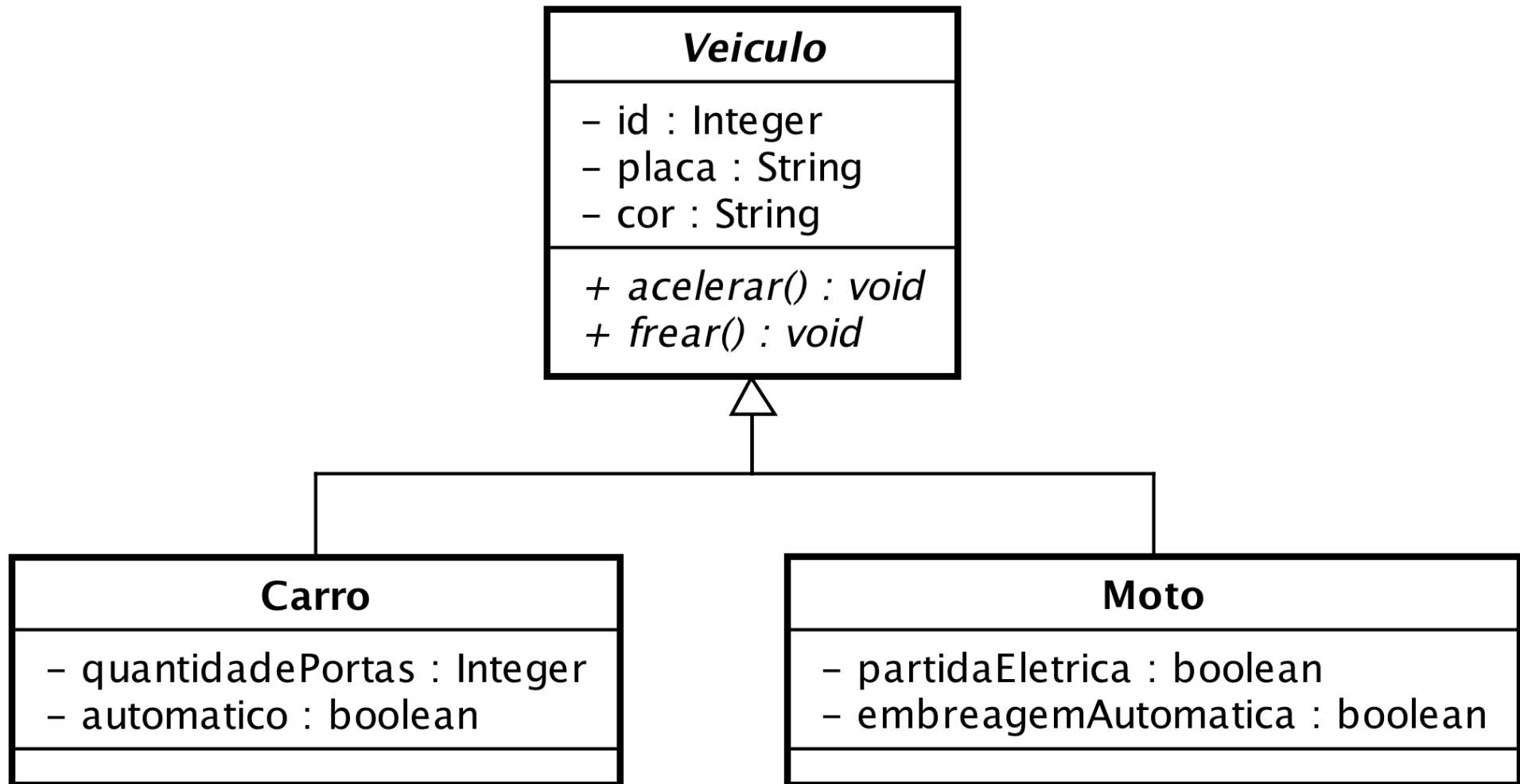
Métodos Abstratos - Conceito

- Um método abstrato não possui corpo, apenas a assinatura do método;
 - Ex. em Java: ***public abstract void acelerar();***
- Igualmente a classe abstrata, em Java, um método abstrato deve utilizar o modificador de acesso ***abstract***;
- A classe concreta que possui um relacionamento de herança com uma classe abstrata, deve implementar (sobrescrever) todos os métodos abstratos;

Métodos Abstratos - Conceito

- Enquanto existir ao menos um método abstrato, a classe deve ser abstrata;
- Em uma classe abstrata pode conter métodos abstratos e concretos;
- Não existe método abstrato privado em Java;
- Em Java não existe atributo abstrato.

Métodos Abstratos – Exemplo em UML



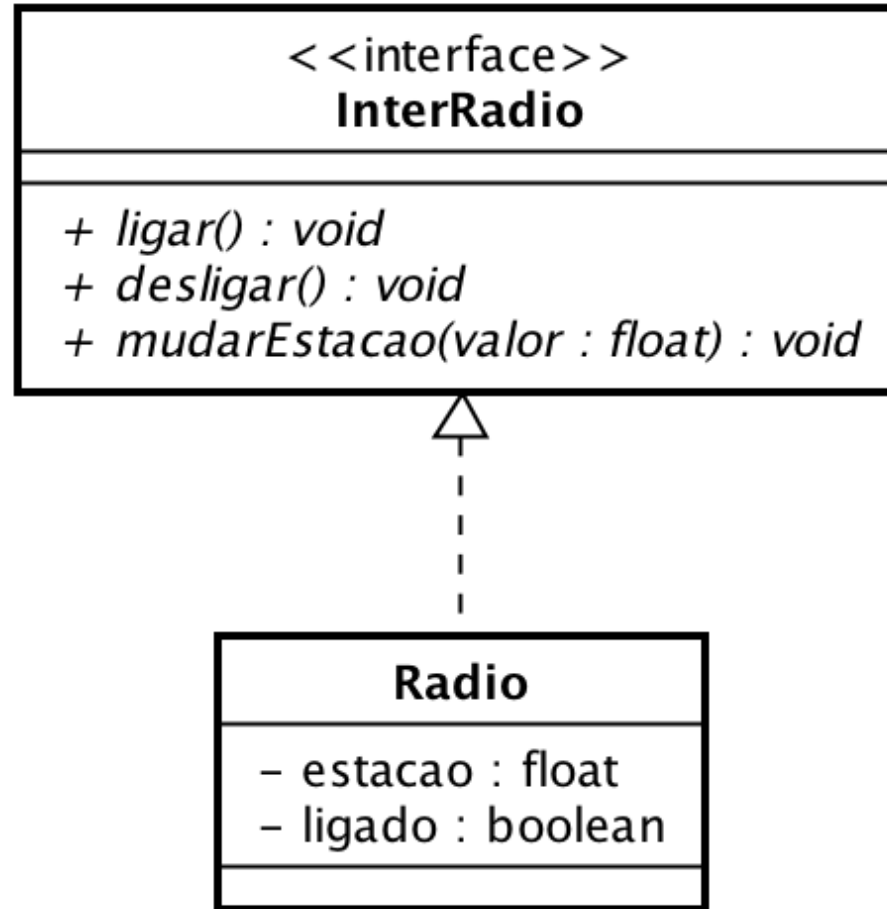


INTERFACE

Interface – Conceito

- Uma interface é utilizada para **definir o comportamento** de uma classe;
 - *Ela estabelece um “**contrato**” a ser seguido pela classe.*
- A classe, que implementa uma interface, **deve disponibilizar (obrigatoriamente)** todo o comportamento definido pela interface.

Interface – Exemplo em UML



Interface – Exemplo em Java

```
public class Radio implements InterRadio {  
    private float estacao;  
    private boolean ligado;  
    @Override  
    public void ligar() {  
        ligado = true;  
    }  
    @Override  
    public void desligar() {  
        ligado = false;  
    }  
    @Override  
    public void mudarEstacao(float valor) {  
        this.estacao = valor;  
    }  
}
```

```
public interface InterRadio {  
    public void ligar();  
    public void desligar();  
    public void mudarEstacao(float valor);  
}
```

Interface (Java)

- Tipo de método: Uma **interface** pode conter ***apenas métodos abstratos**;
 - ❖ *A partir do Java 8, uma interface também pode conter métodos **default** e **static**.*
- Tipo de atributo: Em uma **interface** pode existir apenas atributos do tipo **static** e **final**. Todas as variáveis são do tipo **final** (mesmo não utilizando a *keyword final*);
- Keyword **implements**: Apenas classes podem implementar uma interface. Uma classe pode implementar uma ou mais interfaces;
- Keyword **extends**: Uma interface pode herdar o comportamento (somente) de outra interface. Pode existir herança múltipla em interfaces;
- Acessibilidade: Por padrão os atributos e os métodos são do tipo **public** (mesmo não utilizando a *keyword*).



CLASSE ABSTRATA VS INTERFACE



Quando usar

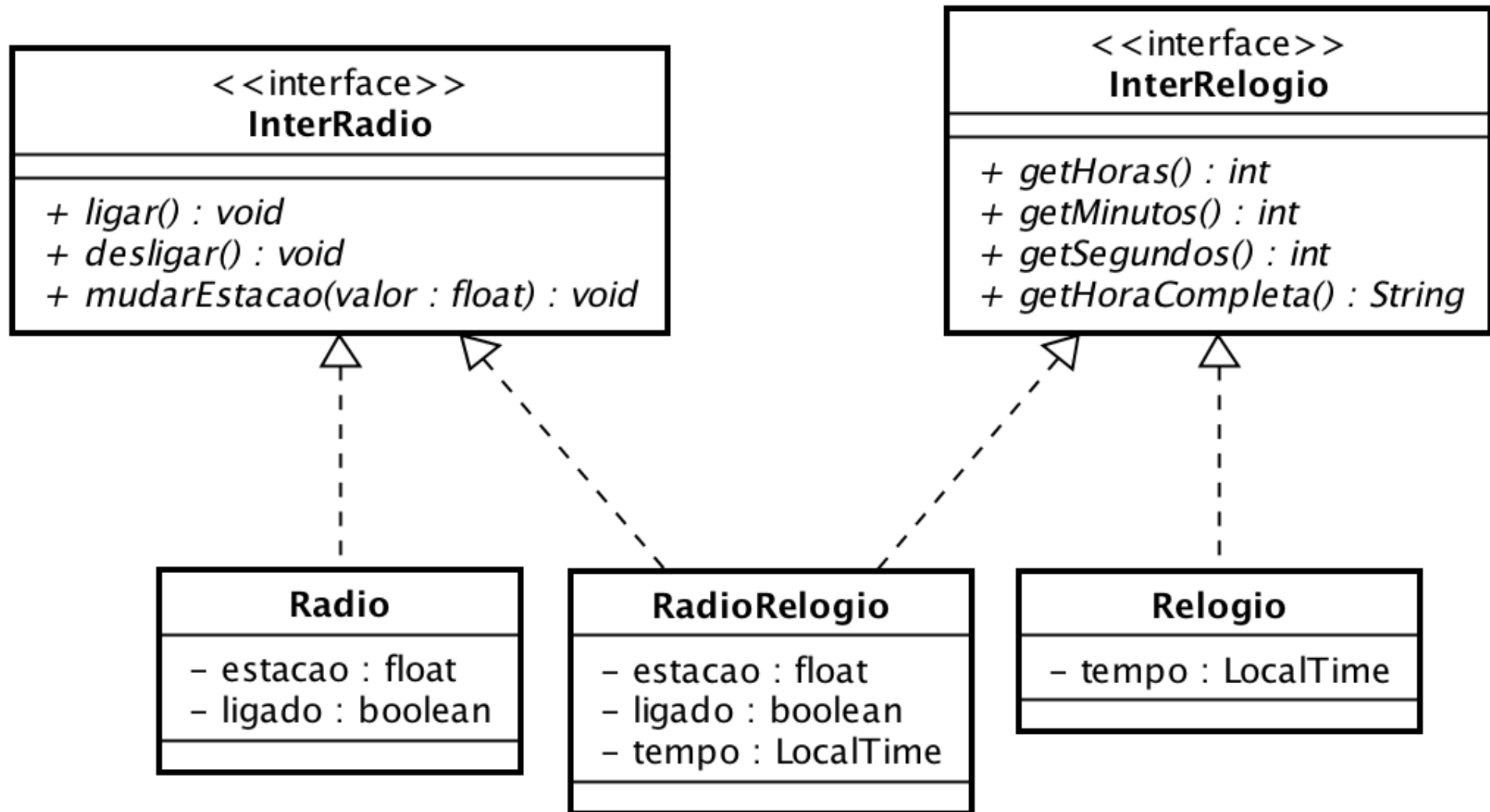
- Classes Abstratas:

- *Necessidade de compartilhamento de código comum entre classes:*
 - Atributos e Métodos;
 - Métodos abstratos e não abstratos;
 - Atributos estáticos e não estáticos;
 - Modificadores de acesso variados.

- Interface:

- *Geralmente todos os recursos são abstratos;*
- *Uma classe pode implementar mais de uma interface;*
- *Para definir o comportamento de um tipo de dado, porém, sem a necessidade de se preocupar com quem implementa.*

Exercício: Implemente o Diagrama abaixo



A thick black L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

ALGORITMOS E PROGRAMAÇÃO II

Obrigado.