



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ  
IFCE *CAMPUS* FORTALEZA  
TELEMÁTICA

Disciplina: **Programação orientada a objeto** (01.302.11)

Professor: **César Olavo**

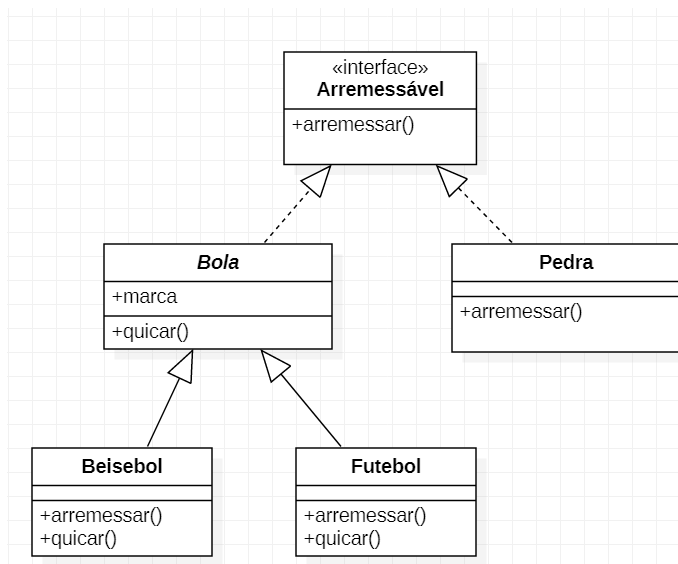
Alunos:

**Cícera Márcia da Fonseca Silva** (mat. 20222013020016)

**Pablo Busatto** (mat. 20221013020042)

### Interfaces Java e UML

Implemente a seguinte hierarquia de classes no editor. Você pode preencher os corpos do método para os métodos arremessar ou quicar com simples `System.out.println`.



```

package arremesso;

public interface Arremessavel {
    void arremessar();
}

```

---

```

package arremesso;

public abstract class Bola implements Arremessavel {
    public String marca;
    public void quicar() {
        System.out.println("Bola quicando.");
    };
}

```

---

```

package arremesso;

public class Pedra implements Arremessavel {

    @Override
    public void arremessar() {
        System.out.println("Pedra arremessada.");
    }
}

```

---

```

package arremesso;

public class Beisebol extends Bola {

    @Override
    public void arremessar() {
        System.out.println("Bola de beisebol arremessada.");
    }

    @Override
    public void quicar() {
        System.out.println("Bola de beisebol quicou.");
    }
}

```

---

```

package arremesso;

public class Futebol extends Bola {

    @Override
    public void arremessar() {
        System.out.println("Bola de futebol arremessada.");
    }

    @Override
    public void quicar() {
        System.out.println("Bola de futebol quicou.");
    }
}

```

---

```

package arremesso;

public class Main {

    public static void main(String[] args) {
        Futebol bolaFut = new Futebol();
        bolaFut.arremessar();
        bolaFut.marca = "Nike";
    }
}

```

Responda as seguintes questões:

Em que classes abstratas e interfaces são iguais?

- Ambas podem ter métodos abstratos.
- Interfaces e classes abstratas não podem ser instanciadas (com operador `new`).
- Assim como de classes abstratas, podem ser criadas variáveis de interface (apesar de não poderem ser instanciadas).

Em que classes abstratas e interfaces são diferentes?

- Classes abstratas podem ter métodos concretos, interfaces não.
- Todos os métodos das interfaces são públicos e abstratos.
- Os métodos das interfaces são implicitamente públicos e abstratos.
- Os campos das interfaces são sempre `public static final`.
- Os campos das interfaces são implicitamente `public static final`.
- Interfaces não possuem atributos de instância.
- Interfaces não implementam métodos.
- Classes podem implementar várias interfaces, mas só podem estender de uma classe abstrata ou concreta.
- Interfaces não são classes.
- Interfaces não podem ter métodos estáticos.

Preencha a tabela a seguir com um dos três valores.

- ✓ Um objeto do tipo indicado pode ser armazenado em uma variável do tipo indicado.
- X Um objeto do tipo indicado não pôde ser armazenado em uma variável do tipo indicado.
- Não é possível instanciar um objeto do tipo indicado.

		Tipo de variável				
		Arremessável	Bola	Pedra	Beisebol	Futebol
Tipo de objeto	Arremessável	—	—	—	—	—
	Bola	—	—	—	—	—
	Pedra	✓	X	✓	X	X
	Beisebol	✓	✓	X	✓	X
	Futebol	✓	✓	X	X	✓

Qual dos seguintes irá **compilar** (marcar com um **C**)? Qual dos seguintes será executado com sucesso (marque com um **E**)?

1. `Bola Bola = new Futebol();`  
Compila (**C**) e executa (**E**).
2. `Bola Bola = new Futebol();`  
`Beisebol beisebol = (Beisebol)bola;`  
Não compila e não executa.

Java diferencia maiúsculas de minúsculas, por isso não reconhece a variável `bola` a ser convertida.

```
Exception in thread "main" java.lang.Error: Unresolved compilation
problem:
    bola cannot be resolved to a variable

    at arremesso.Main.main(Main.java:7)
```

3. `Object obj = new Beisebol();`  
Compila (**C**) e executa (**E**).
4. `Object obj = new Beisebol();`  
`Arremessavel arremessavel = obj`  
Não compila e não executa.

Para ser colocado em uma variável mais especializada, é preciso fazer uma conversão de tipo explícita.

```
Exception in thread "main" java.lang.Error: Unresolved compilation
problem:
    Type mismatch: cannot convert from Object to Arremessavel

    at arremesso.Main.main(Main.java:7)
```

5. `Arremessavel arremessavel = new Beisebol();`  
`Objeto obj = arremessavel;`  
Compila (**C**) e executa (**E**).

Para ser colocado em uma variável mais genérica, **não** é preciso fazer uma conversão de tipo explícita.