

A close-up, high-angle photograph of a black vinyl record on a turntable. The record has a bright red center label. A black tonearm is positioned over the record, with its stylus resting on the grooves. The turntable's platter is visible, showing concentric grooves. The background is dark and out of focus.

Record in Java

MBF

Record : example

Definition

```
public record Person(String name, int age) { }
```

Usages

```
Person p1 = new Person("John", 20);  
Person p2 = new Person("John", 20);  
System.out.println("Test toString : " + p1);  
System.out.println(p1.equals(p2));  
System.out.println(p1 == p2);  
System.out.println("you can ask : " + "name : " + p1.name() + "age : " + p1.age());
```

```
Test toString : Person[name=John, age=20]  
true  
false  
you can ask : name : John age : 20
```

Record : a class type

A record aggregates values in an **immutable** way

- *private, final* field for each piece of data
- getter for each field
- *public* constructor with a corresponding argument for each field
- *equals* method that returns *true* for objects of the same class when all fields match
- *hashCode* method that returns the same value when all fields match
- *toString* method that includes the name of the class and the name of each field and its corresponding value

```
public record Person(String name, int age) {
```

Record : example

```
public Person {  
    if (name == null || name.isBlank())  
        throw new IllegalArgumentException("name must not be null or blank");  
}  
public Person(){  
    this("unknown");  
}  
  
public Person(String s) {  
    this(s, 0);  
}  
  
public String toString(){  
    return "Person["+name+", "+age+"]";  
}
```

Usages

```
Person p = new Person();  
System.out.println(p);  
Person p3 = new Person("John");  
System.out.println(p3);  
p3= new Person(p3.name(), p3.age()+1);  
System.out.println(p3);
```

```
Person[unknown, 0]  
Person[John, 0]  
Person[John, 1]
```

Record : Application

```
private record Pair (int height, int lowness){  
    public Pair() {  
        this(0,0);  
    }  
}
```

Usages

```
if ( height <= 2*lowness )  
    return new Pair(height,lowness);
```

```
private static final Pair LEAF_PAIR = new Pair();
```