

# QCM test Java

**20 questions en anglais avec réponses.**

**~~Mauvaises réponses~~**

**1) What will be the result of the following expression: `Math.sqrt(16) + Math.pow(2, 3) / 2` ?**

- ~~A) 10.0~~
- B) 8.0
- ~~C) 6.0~~
- ~~D) 12.0~~

## 2) Which method is used to replace all occurrences of a substring within a **StringBuilder** object?

- A) `replace()`
- ~~B) `set()`~~
- ~~C) `change()`~~
- ~~D) `alter()`~~

### 3) Which Date-Time API class is used to represent a date without time and time-zone information?

- ~~A) LocalDateTime~~
- ~~B) ZonedDateTime~~
- C) LocalDate
- ~~D) Instant~~

## 4) Which of the following statements about switch expressions in Java SE 21 is true?

- A) Switch expressions can return a value.
- ~~B) Switch expressions do not allow multiple cases for a single value.~~
- ~~C) Switch expressions must end with a semicolon.~~
- ~~D) Switch expressions cannot include a default case.~~

## 5) What is true about an object's eligibility for garbage collection in Java?

- ~~A) An object becomes eligible for garbage collection when it is assigned to a new variable.~~
- B) An object becomes eligible for garbage collection when there are no more references to it.
- ~~C) An object is automatically garbage collected when it goes out of scope.~~
- ~~D) An object must be manually deleted using the delete() method.~~

## 6) Which of the following statements is correct about records in Java?

- ~~A) Records can have instance fields but not static fields.~~
- B) Records are immutable data carriers.
- ~~C) Records do not support constructors.~~
- ~~D) Records cannot implement interfaces.~~



## 7) How can you account for Daylight Saving Time when working with time-zone objects in Java?

- A) Use the ZoneId class to handle time-zone offsets automatically.
- ~~B) Manually adjust the time using LocalDateTime.~~
- ~~C) Daylight Saving Time is automatically ignored by the Date-Time API.~~
- ~~D) Use the Calendar class instead of the Date-Time API.~~

## 8) What will be the output of the following code?

```
int a = 10;  
int b = 3;  
double c = (double) a / b;  
System.out.println(c);
```

- ~~A) 3~~
- ~~B) 3.0~~
- ~~C) 3.3333~~
- D) 3.333333333333333333333335

## 9) Which of the following is a valid switch expression in Java SE 21?

- A) ~~int result = switch (value) { case 1 -> "One"; case 2 -> "Two"; default -> "Unknown"; };~~
- B) `int result = switch (value) { case 1 -> 1; case 2 -> 2; default -> 0; };`
- C) `String result = switch (value) { case 1 -> "One"; case 2 -> "Two"; default -> "Unknown"; };`
- D) ~~All of the above~~

## 10) Which statement about the life-cycle of an object in Java is true?

- A) An object's life begins when its constructor is called and ends when it is garbage collected.
- ~~B) An object's life begins when the program starts and ends when the program ends.~~
- ~~C) An object's life begins when it is assigned to a variable and ends when the variable goes out of scope.~~
- ~~D) An object's life begins and ends within the same method call.~~

# 11) Which of the following code correctly utilize Java Date/Time API for DST?

~~A:~~

```
LocalDate date = LocalDate.now();  
System.out.println(date);
```

~~C:~~

```
Date date = new Date();  
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
sdf.setTimeZone(TimeZone.getTimeZone("Europe/London"));  
System.out.println(sdf.format(date));
```

~~B:~~

```
Calendar cal = Calendar.getInstance();  
TimeZone tz = TimeZone.getTimeZone("Europe/London");  
cal.setTimeZone(tz);
```

D :

```
LocalDateTime localDateTimeBeforeDST = LocalDateTime.now();  
ZoneId uk = ZoneId.of("Europe/London");  
ZonedDateTime zonedDateTimeBeforeDST = localDateTimeBeforeDST.atZone(uk);  
ZonedDateTime zonedDateTimeAfterDST = zonedDateTimeBeforeDST.plus(10, ChronoUnit.DAYS);
```

<https://www.baeldung.com/java-daylight-savings>

## 12) Which of the choices will fail to compile?

A:

```
String myArticle =
    ""
    Java SE 21 is Good.
    This Tests are realy useful
    for latest Java Certifications. " """;
```

C:

```
String code =
    ""
    String text = \"""
    A text block inside a text block
    \""";
    """;
```

~~B:~~

```
String type = ...
String code = String.format("""
public void print(%s o) {
System.out.println(Objects.toString(o));
}
""", type);
```

~~D:~~

```
String myQuote = """ Pain is the healing process.
Failue is the success process.""";
```

<https://www.baeldung.com/java-text-blocks>

## 13) Code :

```
class Exam {  
    public String toString() {  
        return "Practice Exam";  
    }  
}  
  
class PracticeExam {  
  
    public static void main(String args[]) {  
        Object o = new Exam();  
  
        switch (o) {  
            case null -> System.out.println("null");  
            case String s -> System.out.println("String");  
            case Exam e -> System.out.println(e.toString());  
            case int[] ia -> System.out.println("Array length" + ia.length);  
            default -> System.out.println("Something else");  
        }  
    }  
}
```

What is the output?

- ~~A) @Exam~~
- ~~B) String~~
- ~~C) Array length 0~~
- ~~D) null~~
- E) Practice Exam
- ~~F) Something else~~

## 14) Which statement about the sealed modifier in Java SE 21 is correct?

- ~~A) It allows a class to have no subclasses.~~
- B) It specifies which classes can inherit from a class.
- ~~C) It enforces immutability on a class.~~
- ~~D) It is used to create immutable collections.~~



## 15) What is the correct way to create an immutable List using the List factory methods?

- A) `List list = List.of("A", "B", "C");`
- ~~B) `List list = new ArrayList<>(Arrays.asList("A", "B", "C"));`~~
- ~~C) `List list = List.immutable("A", "B", "C");`~~
- ~~D) `List list = Collections.unmodifiableList(Arrays.asList("A", "B", "C"));`~~

## 16) Which feature of records was introduced in Java 21?

- ~~A) Records can implement interfaces.~~
- B) Records now support sealed types.
- ~~C) Records can have mutable fields.~~
- ~~D) Records can be instantiated without defining their components.~~

## 17) Which statement about the `String.strip()` method is correct?

- ~~A) It removes all whitespace from a string.~~
- ~~B) It trims only leading whitespace from a string.~~
- C) It trims both leading and trailing whitespace from a string.
- ~~D) It removes all characters that are not alphanumeric.~~

## 18) What is the purpose of the instanceof pattern matching introduced in Java 21?

- A) To eliminate the need for explicit type casting after an instanceof check.
- ~~B) To improve the performance of instanceof checks.~~
- ~~C) To allow instanceof to match multiple types simultaneously.~~
- ~~D) To simplify the creation of sealed classes.~~

# 19) What is the correct way to use the switch expression with multiple labels in Java 21?

A.

```
switch (day) {
    case "MON", "TUE", "WED" -> System.out.println("Weekday");
    case "SAT", "SUN" -> System.out.println("Weekend");
    default -> System.out.println("Unknown");
}
```

~~B.~~

```
switch (day) {
    case "MON" || "TUE" || "WED" -> System.out.println("Weekday");
    case "SAT" || "SUN" -> System.out.println("Weekend");
    default -> System.out.println("Unknown");
}
```

€.

```
switch (day) {
    "MON", "TUE", "WED" : System.out.println("Weekday");
    "SAT", "SUN" : System.out.println("Weekend");
    default : System.out.println("Unknown");
}
```

~~D.~~

```
switch (day) {
    case ["MON", "TUE", "WED"] -> System.out.println("Weekday");
    case ["SAT", "SUN"] -> System.out.println("Weekend");
    default -> System.out.println("Unknown");
}
```

## 20) What is the main purpose of Virtual Threads introduced in Java 21?

- A) To provide a mechanism for lightweight threads that scale better with high concurrency.
- ~~B) To replace traditional threads with a completely new thread model.~~
- ~~C) To reduce the memory footprint of synchronized blocks.~~
- ~~D) To ensure backward compatibility with older threading models.~~