Exercise walk-through

Wrapper Classes ▪ 4.7

eXercise

You must be able to

- understand that a **wrapper class 包装类** lets a primitive be treated as an object
- use the `Integer` and `Double` classes
- describe **autoboxing 自动装箱** and **unboxing 拆箱**

Method — Wrapper classes

A **wrapper class** lets a primitive value be treated as an **object**: `Integer` wraps an `int`, and `Double` wraps a `double`. This is needed because some structures (like `ArrayList`) store only objects.

Method — Autoboxing and unboxing

```
nInteger+w nn+w o=+w l+m+mi5p;+w   c+c1// autoboxing: int  Integer
k+ktint+w nm+w o=+w nnp;+w       c+c1// unboxing: Integer  int
```

Practice 2.1 ▪ 1 mark

State what a wrapper class does.

Solution 2.1

Method: Wrapper classes

Worked steps

it lets a primitive value be treated as an object B1.

Practice 2.2 ▪ 1 mark

Name the wrapper class for `int`.

Solution 2.2

Method: Wrapper classes

Worked steps

`Integer` B1.

Practice 2.3 ▪ 2 marks

State the difference between autoboxing and unboxing.

Solution 2.3

Method: Autoboxing and unboxing

Worked steps

autoboxing converts a primitive to its wrapper; unboxing converts a wrapper back to a primitive B1 B1.

Exam-style 3.1 ▪ 1 mark

The wrapper class for `double` is

A `Double`

B `double`

C `Number`

D `Float`

Solution 3.1

Method: Wrapper classes

A `Double` ✓

B `double`

C `Number`

D `Float`

Worked steps

A.

Exam-style 3.2 ▪ 1 mark

Converting an `int` to an `Integer` automatically is

A unboxing

B autoboxing

C casting

D overloading

Solution 3.2

A unboxing

B autoboxing ✓

C casting

D overloading

Worked steps

B.

Exam-style 3.3 ▪ 1 mark

`Integer x = 7;`.
(a) Name what happened to the `7`.
(b) Name the wrapper class used.
(c) Name the reverse process (`Integer` to `int`).

Solution 3.3

Method: Wrapper classes

Worked steps

(a) it was autoboxed into an `Integer` B1. (b) `Integer` B1. (c) unboxing B1.