Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

`in`

`tidy`: `after` is skipped

`caught`

2.2

yes: a `TooLongError` is a `LoanError`

no: a plain `LoanError` is not a `TooLongError`

2.3

“look before you leap” (LBYL) and “easier to ask forgiveness” (EAFP)

another process can delete the file between the check and the `open`

with `try`, the one `open` call both checks and uses the file, so there is no gap

2.4

every error is swallowed, including bugs such as a `TypeError`

the order may not be saved and nobody is told

3.1

(a) `parsed`, `cleanup`; returns `25`

(b) `bad`, `cleanup`; returns `-1`

(c) `parsed`, `cleanup`; a `ZeroDivisionError` reaches the caller, because no clause catches it

(d) `finally` runs on every way out of the `try`, including `return`

3.2

(a) `class LibraryError(Exception): pass`; `class LoanError(LibraryError)`; `class NotFoundError(LibraryError)`; `class OverdueError(LoanError)`; `class LimitError(LoanError)`

(b) `except LoanError` and `except LibraryError`; not `except NotFoundError`

(c) the general clause catches it first, so the `LimitError` handler never runs

3.3

(a)

```
def save_with_retry(order):
    for attempt in range(3):
        try:
            return save(order)
        except ConnectionError:
            if attempt == 2:
                raise
```

loop of at most 3 tries; catches only `ConnectionError`; re-raises after the last try; `IntegrityError` is not caught, so it goes up at once

(b) the same data breaks the same rule every time, so a retry cannot succeed

(c) callers handle one domain error and do not depend on which database is used; the page can show a clear message; `from e` keeps the original error for the log, so nothing is lost for debugging

exercises.lol exercises.com.cn