Lab 12 · Part 5 健壮性与真实世界 AI Level 1

真实文本数据处理

从编码和单行解析开始，完成能定位坏行、统计有效记录并导出 CSV 的日志处理程序。

截止：小作业 4 · 12 月 8 日 周二 23:59

本次目标

- 让随机实验在固定条件下可复现
- 用明确的编码和 `with` 读写文件，保留需要的空白
- 用命名分组解析日志，区分格式检查与语义验证
- 让每条输入都有结果或可定位的诊断，并验证 CSV 往返

A 段约 40 分钟，B 段约 35 分钟，C 段约 15 分钟，三段同时开放。代码题和答题卡在小作业 4提交。AI 政策为 Level 1，须说明实际使用或未使用、修改与验证情况。不要提交密码、私钥或会话令牌。

数据与契约

下载本次日志数据，保存为自己的 `lab12/app.log`。该 UTF-8 文件包含 212 条物理行：200 条有效记录、12 条应拒绝的记录，末尾换行不额外算一行。空行和注释也归入坏行，不能无声忽略。生成器使用 Python 3.13、种子 1602；复现还需保持调用过程一致。

正常记录示例：

```
2026-11-05 14:23:45 INFO user=alice action=login duration=1234ms
```

本实验使用以下明确的教学子集：

- 日期为 ASCII `YYYY-MM-DD`，且为真实日期；时间为 ASCII `HH:MM:SS`，时分秒有效。
- 级别限 `INFO/WARN/ERROR`；用户名符合 `[A-Za-z][A-Za-z0-9_]*`；action 为一个或多个 ASCII 小写字母。
- 字段间允许一个或多个空格或 Tab，不允许跨行、额外前后空白或尾随垃圾。
- 时长为 ASCII 非负整数字符串加 `ms`，数值在 0–3000，返回时只保留数字部分。
- 使用 `re.fullmatch` 或等效的完整匹配及命名分组。用 `datetime.date.fromisoformat`、`datetime.time.fromisoformat` 验证日期时间；捕获预期的 `ValueError`，不屏蔽其他程序错误。

A 段 · 编码与单行处理

12-1 编码实验

实现 `encoding_lengths(text: str) -> tuple[int, int]`，返回 UTF-8 和 GBK 的字节长度；GBK 无法表示时让 `UnicodeEncodeError` 正常传给调用者，不替换或忽略字符。

答题卡中附上三组实际代码与输出：`"计算导论"` 的两种长度；`"你好"` 的 UTF-8 字节按 Latin-1 解码及可逆性；相同字节按 ASCII 解码的完整异常。解释为什么“不报错”和“文本正确”是两回事。

12-2 安全地读文件

```
def read_lines(path: str) -> list[str]:
    """逐行读 UTF-8，返回去掉行尾换行的列表。"""
```

使用 `with`，直接遍历文件，不使用 `readlines()`；只删除行尾换行，保留其他空格。空行保留为空字符串，末行无换行也要处理。文件不存在时抛出包含路径的 `FileNotFoundError`，保留异常原因。

答题卡中说明：函数返回完整列表，文件增大时内存为什么仍会增长？若只需计数，可以怎样改变处理方式？

12-3 解析一条日志

```
def parse_line(line: str) -> dict[str, str] | None:
    """输入不含行尾换行；合法时返回六字段字符串字典，否则返回 None。"""
```

六个键为 `date/time/level/user/action/duration`，严格遵守上面的数据契约。用命名分组，先完成所有验证再返回。空串、注释、缺字段、无效日期时间、非 ASCII 数字、越界时长和尾随垃圾都返回 `None`。

准备至少三个正常用例和五个反例。不要只验证能匹配给出的正常行。

12-4 联系方式提取

实现三个函数，保持出现顺序并保留重复项：

```
def extract_emails(text: str) -> list[str]: ...
def extract_phones(text: str) -> list[str]: ...
def email_domains(text: str) -> list[str]: ...
```

这里明确使用教学子集，不验证完整邮箱标准。候选项由空白、中文/英文逗号、分号、冒号或 `/` 分隔。邮箱本地部分为一个或多个 `[A-Za-z0-9._-]`；域名由至少两个非空 `[A-Za-z0-9-]+` 标签以点连接；手机号为 `1` 开头的 11 位 ASCII 数字。对每个候选项完整匹配，不截取一个无效候选项内部的子串。`email_domains` 返回匹配邮箱中 `@` 后的部分。

```
张三: zhangsan@sjtu.edu.cn 13812345678
李四: lisi.wang@example.com, 15900001111
重复: zhangsan@sjtu.edu.cn
无效: not-an-email@ / 123456 / x13812345678y
```

12-5 可复现的掷骰实验

```
def roll_dice(n: int, seed: int) -> list[int]: ...
```

用局部 `random.Random(seed)`，按顺序调用 `randint(1, 6)` 共 n 次，返回结果列表。n 为 0 返回空列表，负数抛出 `ValueError`。每次函数调用初始化一次，不在每次抽样前重置，不改变模块级 `random` 的状态。在相同 Python 版本下相同参数应给出相同结果。

B 段 · 一份日志的完整处理

12-6 加载、诊断与统计

复用已完成的 `parse_line`，在同一份提交文件中实现：

```
def load_log(path: str) -> tuple[
    list[dict[str, str]],
    list[tuple[int, str]],
]: ...
def count_by_level(records: list[dict[str, str]]) -> dict[str, int]: ...
def busiest_user(records: list[dict[str, str]]) -> str: ...
def average_duration(records: list[dict[str, str]]) -> float: ...
```

`load_log` 用 UTF-8 逐行读取，返回成功记录与坏行 `(1起始物理行号，去掉换行后的原文)`，均保持顺序。空行和注释也归入坏行；有效记录不丢弃。读取、权限和解码异常传给调用者，不作为某条坏行吞掉。对坏行给出可定位的说明，至少包含文件名、行号、原文和失败原因；若只定位到格式不符，明确列出预期结构，不伪造字段原因。

统计函数的输入为已经验证的记录，不能再次随意删除记录。`count_by_level` 只返回出现过的级别；`busiest_user` 按次数最多选用户，并列时选字典序最小者，空输入返回 `""`；`average_duration` 以所有有效记录为分母，空输入返回 `0.0`。不要修改传入列表或字典。

在 B 答题卡中报告整份文件的成功数、坏行数、每条坏行的位置和诊断，以及两个计数之和。至少演示一条坏输入到具体诊断的完整过程。保存全量记录是本题的接口要求，不能称为常量内存实现。

12-7 输出与读回 CSV

```
def write_csv(records: list[dict[str, str]], path: str) -> None: ...
```

用 `csv` 模块、UTF-8、`newline=""`，按 `date,time,level,user,action,duration` 的顺序写表头和记录，保持记录顺序。空输入仍写表头；已存在的目标文件允许覆盖，调用前确认这是输出路径。文件操作异常传给调用者。

使用 `csv.DictReader` 读回，断言读回的完整记录列表等于写出的列表，不只比较条数。再用带逗号或引号的字段单独验证 CSV 的引号处理；此格式实验不改变 `parse_line` 的用户名约束。

B-1 失败边界与测试

提交至少五个反例及预期处理结果，涵盖无效日期、空用户名、非数字或越界时长、尾随垃圾、末行无换行。附一次坏输入的完整诊断和 CSV 往返验证输出。解释文件级失败为何不能统一变成空记录列表。

12-8 扩展：生日模拟

保留原有扩展函数 `birthday_collision(n_people, trials, seed=None) -> float`。假定 365 天独立均匀生日，每次实验只初始化一次生成器。比较 20–30 人的估计，并保存两次相同条件的结果。有限次模拟不保证首次超过 50% 的人数总是 23；不要以这个波动结果作唯一正确性判断。本题作为扩展，不额外增加必交代码题。

C 段 · SSH 与 scp

C-1 远程文件处理

使用课程组提供的练习主机和自己的练习账号；下列大写文字是占位符，不要直接照抄。若尚未发放账号，先完成本地脚本与相对路径检查，在答题卡中注明待完成的远程步骤，不虚构运行证据。

```
ssh USER@HOST
mkdir -p cs1602-lab12
exit
scp lab12/app.log USER@HOST:cs1602-lab12/app.log
scp analyze.py USER@HOST:cs1602-lab12/analyze.py
ssh USER@HOST
cd cs1602-lab12
python3 analyze.py
exit
scp USER@HOST:cs1602-lab12/parsed.csv lab12/parsed-remote.csv
```

先让脚本用相对路径读取 `app.log` 并输出 `parsed.csv`。`scp` 复制文件；`ssh` 登录远端执行命令。按课程组提供的信息核对首次连接的主机指纹；不把练习账号密码、私钥或令牌写进报告。C 卡记录脱敏后的命令、远端工作目录和读回验证，注明未完成步骤。

个人作业 3

本周开始“大整数运算”，覆盖第 12–13 周。接口使用字符串，实现加、减、乘、整除与乘方，并附技术报告；具体题面与截止时间以个人作业 3 提交页和课程日程为准。

提交前检查

- A 段五道代码题与 A 答题卡；B 段加载统计、CSV 两道代码题与 B 答题卡。
- 七道代码题的函数名称、返回结构和异常契约均与题面一致。
- 标准数据为 200 条成功、12 条坏行，合计 212；每条坏行可定位。

- CSV 比较完整字段，不只比较条数；没有把诊断混入数据文件。
- C 段记录实际完成情况；AI 使用声明包含实际验证，未泄露账号信息。