

## Лабораторная работа №5

**Выполнил:** Попков Максим Александрович, М8О-105БВ-25

**Тема:** Отладка кодовой базы проекта на Python с помощью средств отладки

### Цель работы

- закрепление навыков работы с отладчиком;
- формирование понимания типовых логических и runtime-ошибок;
- освоение методики поиска, анализа и устранения ошибок;
- развитие умения объяснять причину некорректного поведения программы.

### Ошибка 1 – изменение коллекции при итерации

Место: simulation.py, функция run\_simulation, вспомогательная \_remove

#### Симптом:

вместо удаления одной случайной книги удаляются все. В сценарии run\_simulation(steps=50, seed=1) после шага 32 в библиотеке 4 книги, на шаге 33 выполняется REMOVE и библиотека опустошается; аналогично после шага 47. При этом в логах показывается только последний удаленный объект, из-за этого несоответствие с фактическим состоянием.

#### Как воспроизвести:

запустить run\_simulation(steps=50, seed=1); проверить размер коллекции перед и после события REMOVE (например, вотч len(library) или list(library) в отладчике) – падает до 0 за одно событие.

#### Отладка:

breakpoint на строке for b in library:. В Locals видно, что цикл идёт по текущей коллекции, на каждой итерации вызывается library.remove\_book(b), коллекция уменьшается, в итоге удаляются все элементы; переменная b после цикла – только последний удалённый, именно его и логирует.

#### Причина:

модификация коллекции, по которой идёт итерация: удаление в for b in library приводит к тому, что за одно событие удаляются все элементы, а лог вводит в заблуждение, показывая только последний.

#### Исправление:

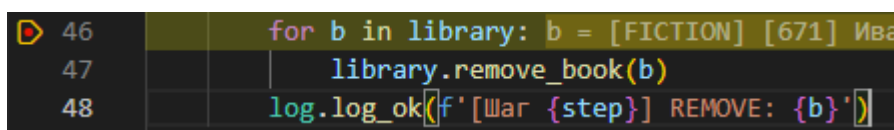
не менять коллекцию, по которой итерируешься: выбрать один случайный элемент из копии (book = random.choice(list(library))), удалить его и логировать; либо итерироваться по копии, если нужно удалить несколько, и логировать каждый.

#### Проверка:

после фикса при том же seed=1 событие REMOVE уменьшает размер библиотеки на 1 (было 4 → стало 3 после шага 33, и т.д.), лог соответствует фактическому состоянию.

#### Доказательства:

До исправления:



```
46 for b in library: b = [FICTION] [671] Ива
47 library.remove_book(b)
48 log.log_ok(f'[Шаг {step}] REMOVE: {b}')
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text)
→ len(sim.library)
Traceback (most recent call last):
  File "<string>", line 1, in <module>
NameError: name 'sim' is not defined. Did you mean: 'sum'?
→ len(library)
4
→ len(library)
0
```

```
[2025-12-19 23:22:19] [INFO] [Шаг 31] FIND_BY_AUTHOR [Эрих Мария Ремарк]: [[FICTION] [1135] Эрих Мария Ремарк - "Трудно быть богом" (жанр: фэнтези, ISBN: 3867565088995) (возрастное ограничение: 19+)]
[2025-12-19 23:22:19] [INFO] [Шаг 32] FIND_BY_AUTHOR [Михаил Булгаков]: [[FICTION] [1196] Михаил Булгаков - "Мастер и Маргарита" (жанр: фэнтези, ISBN: 21840-1107043) (возрастное ограничение: 3+)]
[2025-12-19 23:24:23] [INFO] [Шаг 33] REMOVE: [FICTION] [1196] Михаил Булгаков - "Мастер и Маргарита" (жанр: фэнтези, ISBN: 21840-1107043) (возрастное ограничение: 3+)
[2025-12-19 23:24:23] [INFO] [Шаг 34] ADD: [862] Аркадий Стругацкий - "Ася" (жанр: драма, ISBN: 3414839603062)
```

```
▼ Locals
  ▼ library = <src.library.Library object at 0x00000174A2A6D940>
    > special variables
    > function variables
    ▼ books = <src.collections.BookCollection object at 0x00000174A2A6DA90>
      > special variables
      > function variables
      ▼ _books = [[862] Аркадий Стругацкий - "Ася" (жанр: драма, ISBN: 3414839603062), ...
        > special variables
        > function variables
        > 0 = [862] Аркадий Стругацкий - "Ася" (жанр: драма, ISBN: 3414839603062)
        > 1 = [SCIENCE] [1292] Эрих Мария Ремарк - "Три товарища" (жанр: учебник, ISBN: 6...
        > 2 = [943] Иван Тургенев - "Гарри Поттер и философский камень" (жанр: фэнтези, I...
        > 3 = [FICTION] [671] Иван Тургенев - "Преступление и наказание" (жанр: научпоп, ...
        len() = 4
```

```
▼ Locals
  > b = [FICTION] [671] Иван Тургенев - "Преступление и наказание" (жанр: научпоп, ISBN: 3414839603062)
  ▼ library = <src.library.Library object at 0x00000174A2A6D940>
    > special variables
    > function variables
    ▼ books = <src.collections.BookCollection object at 0x00000174A2A6DA90>
      > special variables
      > function variables
      ▼ _books = []
        > special variables
        > function variables
        len() = 0
```

После исправления:

```
45 | book = random.choice(list(library))
● 46 | library.remove_book(book)
47 | log.log_ok(f'[Шаг {step}] REMOVE: {book}')
```

```
▼ Locals
  ▼ library = <src.library.Library object at 0x000001C52A45D940>
    > special variables
    > function variables
  ▼ books = <src.collections.BookCollection object at 0x000001C52A45DA90>
    > special variables
    > function variables
  ▼ _books = [[FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 360...
    > special variables
    > function variables
    > 0 = [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 360837783...
    > 1 = [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2-4158...
    > 2 = [FICTION] [1135] Эрих Мария Ремарк - "Трудно быть богом" (жанр: фэнтези, IS...
    > 3 = [FICTION] [1196] Михаил Булгаков - "Мастер и Маргарита" (жанр: фэнтези, ISB...
    len() = 4
```

```
> book = [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 360837783...
▼ library = <src.library.Library object at 0x000001C52A45D940>
  > special variables
  > function variables
▼ books = <src.collections.BookCollection object at 0x000001C52A45DA90>
  > special variables
  > function variables
▼ _books = [[FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2...
  > special variables
  > function variables
  > 0 = [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2-4158...
  > 1 = [FICTION] [1135] Эрих Мария Ремарк - "Трудно быть богом" (жанр: фэнтези, IS...
  > 2 = [FICTION] [1196] Михаил Булгаков - "Мастер и Маргарита" (жанр: фэнтези, ISB...
  len() = 3
```

```
[2025-12-19 23:31:00] [INFO] [Шаг 33] REMOVE: [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)
```

## Ошибка 2 – ошибка границы цикла (off-by-one)

Место: simulation.py, функция run\_simulation

Симптом:

при заданных steps выполняется на один шаг больше событий, чем ожидается (для 12 шагов выполняется 13).

Как воспроизвести:

запустить run\_simulation(steps=12, seed=1) под отладчиком или просто посмотреть лог – видно на один шаг больше записей.

Отладка:

breakpoint на строке цикла. В Locals на последней итерации steps=12, step=12, хотя должно быть максимум 11.

Причина:

ошибка границы цикла (off-by-one) – используется range(steps + 1) вместо range(steps).

Исправление:

заменить строку на for step in range(steps):

Проверка:

после фикса step проходит 0..11 при steps=12, количество событий в логге совпадает с 12, итоговое состояние библиотеки соответствует ожиданию.

Доказательства:

До исправления:

```
128 for step in range(steps + 1): step = 12, steps = 12
129 random.choice(commands)()
```

```
✓ VARIABLES
  ✓ Locals
    > commands = [<function run_simulation.<lo...
    > library = <src.library.Library object at...
    seed = 1
    step = 12
    steps = 12
    > _add = <function run_simulation.<locals>...
    > _find_by_author = <function run_simulati...
    > _find_by_genre = <function run_simulatio...
    > _find_by_isbn = <function run_simulation...
    > _find_by_year = <function run_simulation...
    > _find_invalid_isbn = <function run_simul...
    > _find_invalid_year = <function run_simul...
    > _remove = <function run_simulation.<loca...
  ✓ Globals
    > special variables
    > function variables
    > class variables
    > AUTHORS = ['Лев Толстой', 'Фёдор Достоев...
    > FIELDS = ['math', 'physics', 'chemistry'...
    > GENRES = ['роман', 'фантастика', 'драма'...
```

```
[2025-12-19 22:13:17] [INFO] [Шаг 12] FIND_BY_AUTHOR [Лев Толстой]: [[FICTION] [1405] Лев Толстой
- "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)]
```

После исправления:

```
128 for step in range(steps): step = 11, steps = 12
129 random.choice(commands)()
```

```
✓ Locals
  > commands = [<function run_simulation.<lo...
  > library = <src.library.Library object at...
  seed = 1
  step = 11
  steps = 12
  > _add = <function run_simulation.<locals>...
  > _find_by_author = <function run_simulati...
  > _find_by_genre = <function run_simulatio...
  > _find_by_isbn = <function run_simulation...
  > _find_by_year = <function run_simulation...
  > _find_invalid_isbn = <function run_simul...
  > _find_invalid_year = <function run_simul...
  > _remove = <function run_simulation.<loca...
```

```
[2025-12-19 22:25:02] [INFO] [Шаг 11] FIND_BY_GENRE [история]: []
[2025-12-19 22:25:53] [INFO] ИТОГ: в библиотеке 2 книг(и).
[2025-12-19 22:25:53] [INFO] ==== КОНЕЦ СИМУЛЯЦИИ ====
```

Ошибка 3 – неверное условие поиска жанра (find by genre)

Место: library.py, метод find\_by\_genre

Симптом:

поиск по жанру всегда возвращает пустой список, даже если книги с таким жанром есть.

Как воспроизвести:

run\_simulation(steps=12, seed=1); в логах видно, что на 9 шаге добавляется книга жанра «история», на 11 шаге вызов FIND\_BY\_GENRE «история» выдает пустой список.

Отладка:

breakpoint на условии в find\_by\_genre. В Locals при вызове: genre=«история», book.genre=«история», но сравнение идёт с genre.upper() – условие False, список не заполняется.

Причина:

неверное логическое условие — сравнение в разных регистрах (book.genre в lower, genre.upper() в upper).

Исправление:

сравнивать в одном регистре: genre = genre.strip().lower() и if book.genre == genre:

Проверка:

после фикса повторный запуск run\_simulation(steps=12, seed=1) даёт непустой результат для FIND\_BY\_GENRE «история»; локально library.find\_by\_genre(«история») возвращает книгу

Доказательства:

До исправления:

```
89         genre = genre.strip().lower()
90         res = []
91         for book in self.books:
92             if book.genre == genre.upper():
93                 res.append(book)
94         return res
```

```
[2025-12-19 22:28:14] [INFO] [Шаг 9] ADD: [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: исто
рия, ISBN: 36083778353-3) (возрастное ограничение: 9+)
[2025-12-19 22:28:14] [INFO] [Шаг 10] ADD: [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр:
учебник, ISBN: 2-415868-3449) (возрастное ограничение: 16+)
[2025-12-19 22:37:11] [INFO] [Шаг 11] FIND_BY_GENRE [история]: []
```

```
▼ Locals
> book = [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)
  genre = 'история'
> res = []
> self = <src.library.Library object at 0x0000017873381940>
```

Перешли к следующей книге, res по прежнему пустой:

```
▼ Locals
> book = [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2-415868-3449) (возрастное ограничение: 16+)
  genre = 'история'
> res = []
> self = <src.library.Library object at 0x0000017873381940>
```

После исправления:

```

▼ Locals
> book = [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2-415868-3449) (возрастное ограничение: 16+)
genre = 'история'
> res = [[FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)]
> self = <src.library.Library object at 0x00002087405D940>

```

```

[2025-12-19 22:45:40] [INFO] [Шаг 9] ADD: [FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)
[2025-12-19 22:45:40] [INFO] [Шаг 10] ADD: [FICTION] [204] Джоан Роулинг - "Три товарища" (жанр: учебник, ISBN: 2-415868-3449) (возрастное ограничение: 16+)
[2025-12-19 22:46:42] [INFO] [Шаг 11] FIND_BY_GENRE [история]: [[FICTION] [1405] Лев Толстой - "Чистый код" (жанр: история, ISBN: 36083778353-3) (возрастное ограничение: 9+)]

```

```

89     genre = genre.strip().lower()
90     res = []
91     for book in self.books:
92         if book.genre == genre:
93             res.append(book)
94     return res

```

## Ошибка 4 – mutable default

Место: collections.py, конструктор BookCollection.\_\_init\_\_

Симптом:

разные экземпляры BookCollection делятся книгами: добавление в один появляется в другом.

Как воспроизвести:

в Debug Console: создать c1 = BookCollection() и c2 = BookCollection(), добавить книгу в c1, затем проверить len(c2) – станет 1.

Отладка:

```

from src.collections import BookCollection

from src.models import Book

b = Book("title","author",2020,"gence","isbn")

c1 = BookCollection()

c2 = BookCollection()

c1.add_book(b)

print(len(c1), len(c2), id(c1._books), id(c2._books))

```

Видно одинаковые len и одинаковые id списка.

Причина:

использовано изменяемое значение по умолчанию (общий список для всех экземпляров).

Исправление:

вернуть сигнатуру books: list[Book] | None = None, внутри создавать новый список.

Проверка:

после фикса len(c2) остаётся 0, id разных списков отличаются.

Доказательства:

До исправления:

```

14     def __init__(self, books: list[Book] = []) -> None:
15         """
16         Инициализация коллекции. Ничего не возвращает.
17         :param books: Список объектов Book или None для создания пустой коллекции
18         """
19         if type(books) is not list:
20             raise ValueError('Коллекцию можно создать только из списка книг.')
21         self._books = books

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Filter (e.g. text, !exclude, \
→ from src.collections import BookCollection
from src.models import Book
b = Book("title","author",2020,"genge","isbn")
c1 = BookCollection()
c2 = BookCollection()
c1.add_book(b)
print(len(c1), len(c2), id(c1._books), id(c2._books))
1 1 2050266458048 2050266458048

```

После исправления:

```

14     def __init__(self, books: list[Book] | None = None) -> None:
15         """
16         Инициализация коллекции. Ничего не возвращает.
17         :param books: Список объектов Book или None для создания пустой коллекции
18         """
19         self._books = []
20         if books is None:
21             return
22         if type(books) is not list:
23             raise ValueError('Коллекцию можно создать только из списка книг.')
24         for book in books:
25             self.add_book(book)

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
→ from src.collections import BookCollection
from src.models import Book
b = Book("title","author",2020,"genge","isbn")
c1 = BookCollection()
c2 = BookCollection()
c1.add_book(b)
print(len(c1), len(c2), id(c1._books), id(c2._books))
1 0 1710696262272 1710696395456

```

**Ошибка 5 – сравнение через is вместо ==**

Место: models.py, метод Book.\_\_eq\_\_

Симптом:

два разных объекта Book с одинаковыми данными сравниваются как неравные (False).

### Как воспроизвести:

в Debug Console создать два объекта с одинаковыми значениями, избегая интернирования строк:

```
from src.models import Book

b1 = Book("Title 123456", "Author 123456", 2020, "genre", "123-456")

b2 = Book("Title " + "123456", "Author " + "123456", 2020, "genre", "123-456")

print(b1 == b2)
```

Результат False.

### Отладка:

breakpoint в `__eq__` и сравнить Locals: поля одинаковые, но `is` даёт False, потому что это разные объекты строк.

### Причина:

используется сравнение идентичности (`is`) вместо равенства (`==`) для строк/чисел.

### Исправление:

заменить все `is` на `==` в `__eq__`.

### Проверка:

после фикса тот же тест возвращает True.

### Доказательства:

До исправления:

```
45     def __eq__(self, other: "Book") -> bool:
46         """
47         Сравнение двух книг по всем полям.
48         :param other: Объект, которым выполняется сравнение.
49         :return: True, если книги совпадают, иначе False.
50         """
51         if not isinstance(other, Book):
52             return NotImplemented
53         return (self.isbn is other.isbn and self.year is other.year and self.author is other.author and
54                 self.title is other.title and self.genre is other.genre)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Filter (e.g. text, !
→ from src.models import Book
   b1 = Book("Title 123456", "Author 123456", 2020, "genre", "123-456")
   b2 = Book("Title " + "123456", "Author " + "123456", 2020, "genre", "123-456")
   print(b1 == b2)
   False
```



```
▼ Locals
  ▼ other = [2020] Author 123456 - "Title 123456" (жанр: genrex, ISBN: 123-456)
    > special variables
      author = 'Author 123456'
      genre = 'genrex'
      isbn = '123-456'
      title = 'Title 123456'
      year = 2020
  ▼ self = [2020] Author 123456 - "Title 123456" (жанр: genrex, ISBN: 123-456)
    > special variables
      author = 'Author 123456'
      genre = 'genrex'
      isbn = '123-456'
      title = 'Title 123456'
      year = 2020
```

После исправления:

```
45 def __eq__(self, other: "Book") -> bool:
46     """
47     Сравнение двух книг по всем полям.
48     :param other: Объект, с которым выполняется сравнение.
49     :return: True, если книги совпадают, иначе False.
50     """
51     if not isinstance(other, Book):
52         return NotImplemented
53     return (self.isbn == other.isbn and self.year == other.year and self.author == other.author and
54             self.title == other.title and self.genre == other.genre)
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
→ from src.models import Book
b1 = Book("Title 123456", "Author 123456", 2020, "genre", "123-456")
b2 = Book("Title " + "123456", "Author " + "123456", 2020, "genre", "123-456")
print(b1 == b2)
True
```

## Вывод

За время работы выявил и исправил пять логических/runtime-ошибок: off-by-one в цикле симуляции, неверное сравнение жанра, удаление элементов при итерации по той же коллекции, изменяемое значение по умолчанию в BookCollection, сравнение строк через is в \_\_eq\_\_. Каждую ошибку воспроизвёл (фиксированные seed, точки останова), зафиксировал причину и исправление, подтвердил корректность повторным запуском симуляции/проверкой состояния коллекций. Итог: поведение симуляции и поисков соответствует ожиданию, коллекции и сравнение объектов работают корректно, набор типовых багов устранён.