



Практика: Класс `LibraryCard` — Карточка читателя библиотеки

Цель: Закрепить ООП — создать класс с атрибутами, методами и простой логикой. Научиться работать с объектами, списками и форматированным выводом.



Время выполнения: 60 минут



Сложность: средняя (упрощённая) — подходит после 1–2 лекций



Темы: классы, объекты, атрибуты, методы, списки, строки, циклы, условные операторы



Что делаем?

Вы создадите класс `LibraryCard`, который представляет **читательскую карточку**. Карточка хранит:

- Имя читателя
- Номер карточки
- Список книг, которые читатель **взял** (и не вернул)
- Лимит книг (по умолчанию — 3 книги)

Класс должен уметь:

- Брать книгу (если не превышен лимит)
- Возвращать книгу
- Показывать список книг на руках
- Выводить красивый отчёт о состоянии карточки

⚠ Нет дат, нет просрочек, нет сложной валидации — только суть ООП.



Часть 1: Создание класса `LibraryCard` (25 минут)



Атрибуты (в `__init__`):

- `reader_name` — строка (например, "Анна Петрова")
- `card_number` — строка (например, "LC-1001")
- `book_limit` — целое число (по умолчанию 3)
- `borrowed_books` — список строк (названия книг, например: ["1984", "Мастер и Маргарита"])



Пример создания объекта:

```
card = LibraryCard("Анна Петрова", "LC-1001")
```



Методы:

Метод	Описание
-------	----------

Метод	Описание
<code>borrow_book(title)</code>	Добавляет книгу в список, если не превышен лимит. Возвращает <code>True</code> , если удалось, иначе <code>False</code> .
<code>return_book(title)</code>	Удаляет книгу из списка (если она там есть). Возвращает <code>True</code> , если книга найдена и удалена, иначе <code>False</code> .
<code>get_borrowed_books()</code>	Возвращает список всех книг на руках (не возвращённых).
<code>get_status_report()</code>	Возвращает форматированную строку с отчётом о карточке (см. пример ниже).

🔗 Пример отчёта:

```
=== КАРТОЧКА LC-1001 ===
Читатель: Анна Петрова
Лимит: 3 книги
На руках: 2 книги

Список книг:
- 1984
- Мастер и Маргарита
```

🖥️ Часть 2: Интерактивный симулятор (25 минут)

Напишите программу, которая:

1. Создаёт объект `LibraryCard` с именем "Анна Петрова" и номером "LC-1001".
2. Показывает меню:

```
=== Управление карточкой ===
1. Взять книгу
2. Вернуть книгу
3. Показать книги на руках
4. Показать полный отчёт
0. Выход
```

3. При выборе:

- **1** — запрашивает название книги → вызывает `borrow_book()`.
- **2** — запрашивает название книги → вызывает `return_book()`.
- **3** — выводит список книг через `get_borrowed_books()`.
- **4** — выводит отчёт через `get_status_report()`.

🔗 Пример сессии:

Выберите действие: 1
Название книги: Гарри Поттер
→ Книга успешно взята!

Выберите действие: 1
Название книги: Властелин Колец
→ Книга успешно взята!

Выберите действие: 4
=== КАРТОЧКА LC-1001 ===
Читатель: Анна Петрова
Лимит: 3 книги
На руках: 2 книги

Список книг:
- Гарри Поттер
- Властелин Колец

Выберите действие: 0
До свидания!

💡 Если попытаться взять больше книг, чем лимит — выводите:

→ Нельзя взять книгу: лимит исчерпан (3/3)

💡 Если книга уже взята — не добавляйте её второй раз (проверяйте наличие в списке).

✿ Часть 3: Бонусные задания (10 минут)

✅ Бонус 1: Автоматическая нумерация карточек

Добавьте **статический атрибут класса** `next_id = 1001`.

Если при создании карточки `card_number` не указан — генерируйте автоматически: `"LC-" + str(next_id)`, и увеличивайте `next_id` на 1.

🔗 Пример:

```
card1 = LibraryCard("Анна")           # → LC-1001  
card2 = LibraryCard("Борис")          # → LC-1002
```

✅ Бонус 2: Подсчёт общего количества карточек

Добавьте статический атрибут `total_cards`, который увеличивается при создании каждой карточки.

Добавьте **метод класса** `get_total_cards()`, который возвращает это число.

✅ Бонус 3: Метод `has_book(title)`

Возвращает `True`, если книга с таким названием есть в списке `borrowed_books`, иначе `False`.

Примеры входных и выходных данных

Вход:

1. Взять книгу: "Гарри Поттер"
2. Взять книгу: "Властелин Колец"
3. Показать отчёт

Выход:

```
=== КАРТОЧКА LC-1001 ===
Читатель: Анна Петрова
Лимит: 3 книги
На руках: 2 книг

Список книг:
- Гарри Поттер
- Властелин Колец
```

Дополнительно (если закончили раньше)

- Добавьте метод `clear_books()`, который удаляет все книги.
- Реализуйте `__str__()` для краткого вывода: `Анна Петрова (LC-1001): 2/3 книг`
- Добавьте поддержку жанров: храните книги как словари `{"title": "...", "genre": "..."}.`