

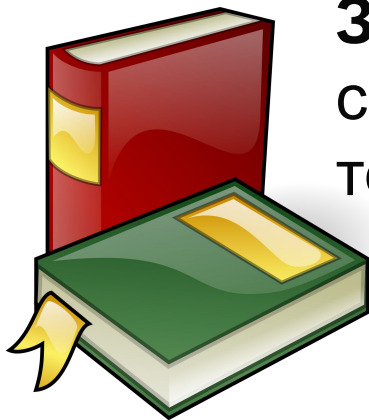
План модуля:

- ▶ Повторение
 - Свойства
 - Методы
 - self
- ▶ Композиция





Начнем с постановки задачи



Задача: написать программу для создания каталога книг различных тематик

Какую информацию хотим хранить по каждой книге?



- Название
- Автор
- Год публикации
- Количество страниц

Создаем класс



- Название
- Автор
- Год публикации
- Количество страниц

```
class Book:
    def __init__(self, name, author, year, pages):
        self.name = name
        self.author = author
        self.year = year
        self.pages = pages
```



Создаем книги



- Название
- Автор
- Год публикации
- Количество страниц

```
class Book:
    def __init__(self, name, author, year, pages):
        self.name = name
        self.author = author
        self.year = year
        self.num_pages = pages
```

```
book1 = Book("Мастер и Маргарита", "Михаил Булгаков", 1967, 480)
book2 = Book("Собачье сердце", "Михаил Булгаков", 1987, 352)
book3 = Book("Сияние", "Стивен Кинг", 2014, 544)
```



```
class Book:
    def __init__(self, name, author, year, pages):
        self.name = name
        self.author = author
        self.year = year
        self.pages = pages
```

```
book1 = Book("Мастер", "М.Булгаков", 1967, 480)
book3 = Book("Сияние", "С.Кинг", 2014, 544)
```

book1

| name | author |
|--------|------------|
| Мастер | М.Булгаков |
| year | pages |
| 1967 | 480 |

book3

| name | author |
|--------|--------|
| Сияние | С.Кинг |
| year | pages |
| 2014 | 544 |



Добавляем метод

Хотим выводить книги в каталоге

```
class Book:
    def __init__(self, name, author, publish_year, num_pages):
        self.name = name
        self.author = author
        self.publish_year = publish_year
        self.num_pages = num_pages

    def to_str(self):
        return f'"{self.name}" author:{self.author}'
```



Используем метод метод

Хотим выводить книги в каталоге

```
class Book:
    def __init__(self, name, author, publish_year,
num_pages):
        self.name = name
        self.author = author
        self.publish_year = publish_year
        self.num_pages = num_pages

    def to_str(self):
        return f'"{self.name}" author:{self.author}'

book1 = Book("Мастер и Маргарита", "Михаил Булгаков", 1967, 480)

print(book1.to_str()) # "Мастер и Маргарита" author:Михаил Булгаков
```



Чтение свойств

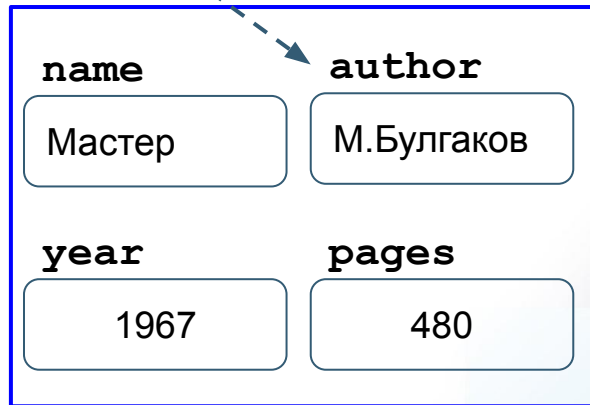
Хотим узнать автора книги

```
class Book:
    def __init__(self, ...):
        self.name = name
        self.author = author
        self.year = year
        self.num_pages = pages
```

```
book1 = Book("Мастер и Маргарита", "Михаил Булгаков", 1967, 480)
```

```
print(book1.author)  # → Михаил Булгаков
```

book1



Чтение свойств

Хотим узнать фамилию автора книги

```
class Book:
    def __init__(self, ...):
        self.name = name
        self.author = author
        self.year = year
        self.num_pages = pages
```

```
book1 = Book("Мастер и Маргарита", "Михаил Булгаков", 1967, 480)
```

```
print(???) # → Булгаков
```

book1

name

Мастер

author

Михаил Булгаков

year

1967

pages

480



Композиция

```
class Author:
    def __init__(self, name, surname):
        self.name = name
        self.surname = surname

class Book:
    def __init__(self, name, author, year, pages):
        self.name = name
        self.author = author
        self.year = year
        self.pages = pages
```

```
author = Author("Михаил", "Булгаков")
book = Book("Вьюга", author, 1926, 25)
```

author

| | |
|--------|----------|
| name | surname |
| Михаил | Булгаков |

book

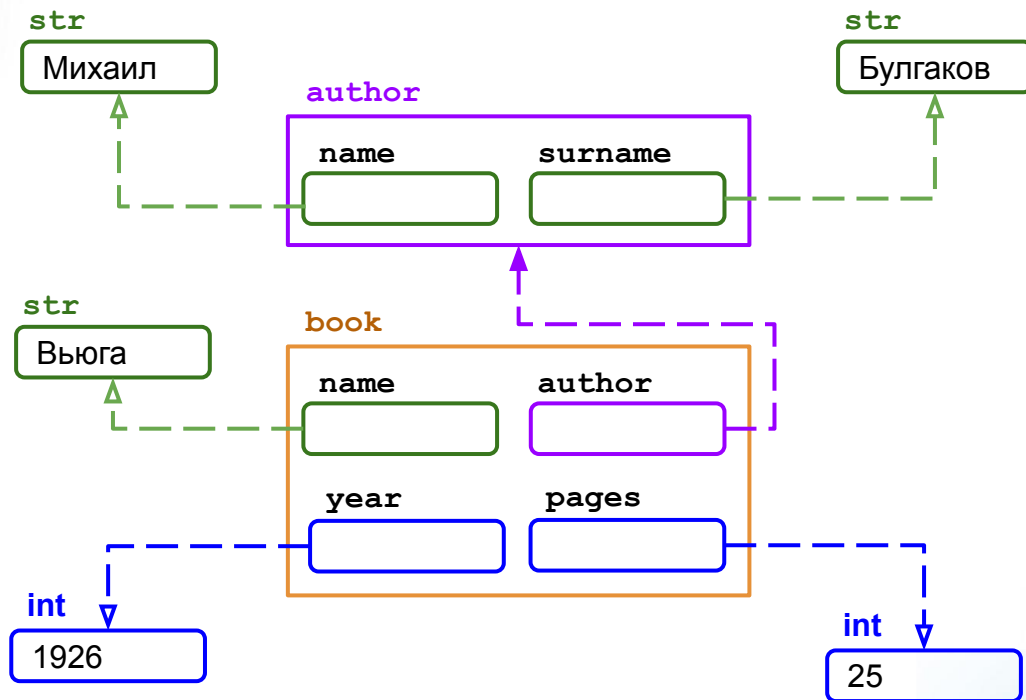
| | |
|-------|--------|
| name | author |
| Вьюга | |
| year | pages |
| 1926 | 25 |



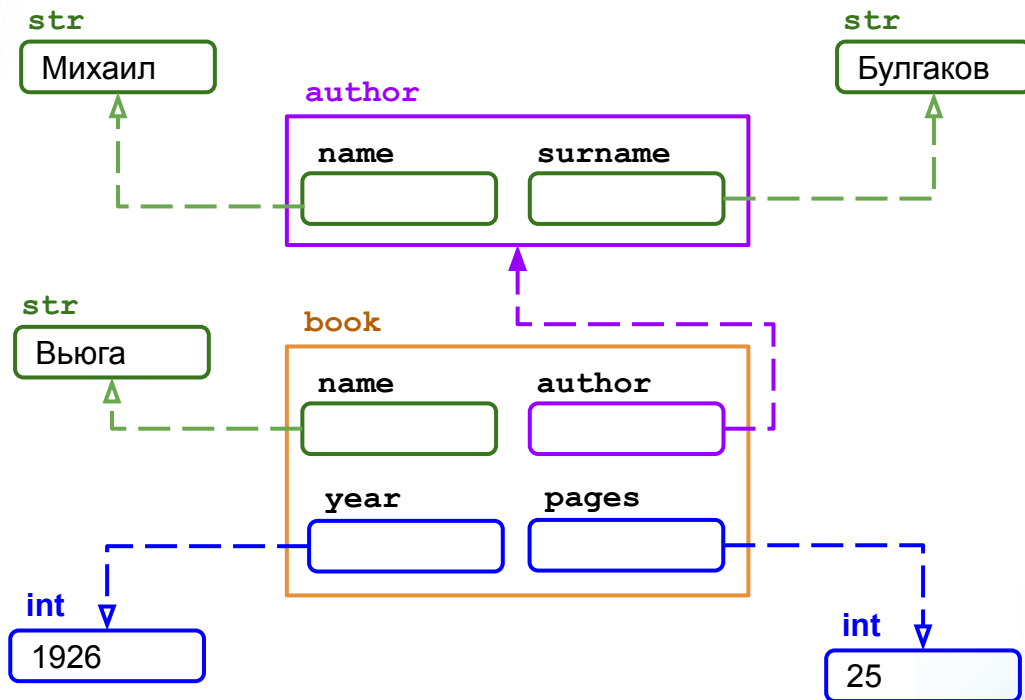
Строки и числа тоже являются объектами



Правильное представление



Правильное представление



```
author = Author("Михаил", "Булгаков")  
book = Book("Вьюга", author, 1926, 25)
```



Узнать фамилию автора книги? ЛЕГКО!

```
class Author:
    def __init__(self, name, surname):
        self.name = name
        self.surname = surname

class Book:
    def __init__(self, name, author, year,
pages):
        self.name = name
        self.author = author
        self.year = year
        self.pages = pages

author = Author("Михаил", "Булгаков")
book = Book("Вьюга", author, 1926, 25)

print(book.author.surname) # → Булгаков
```

