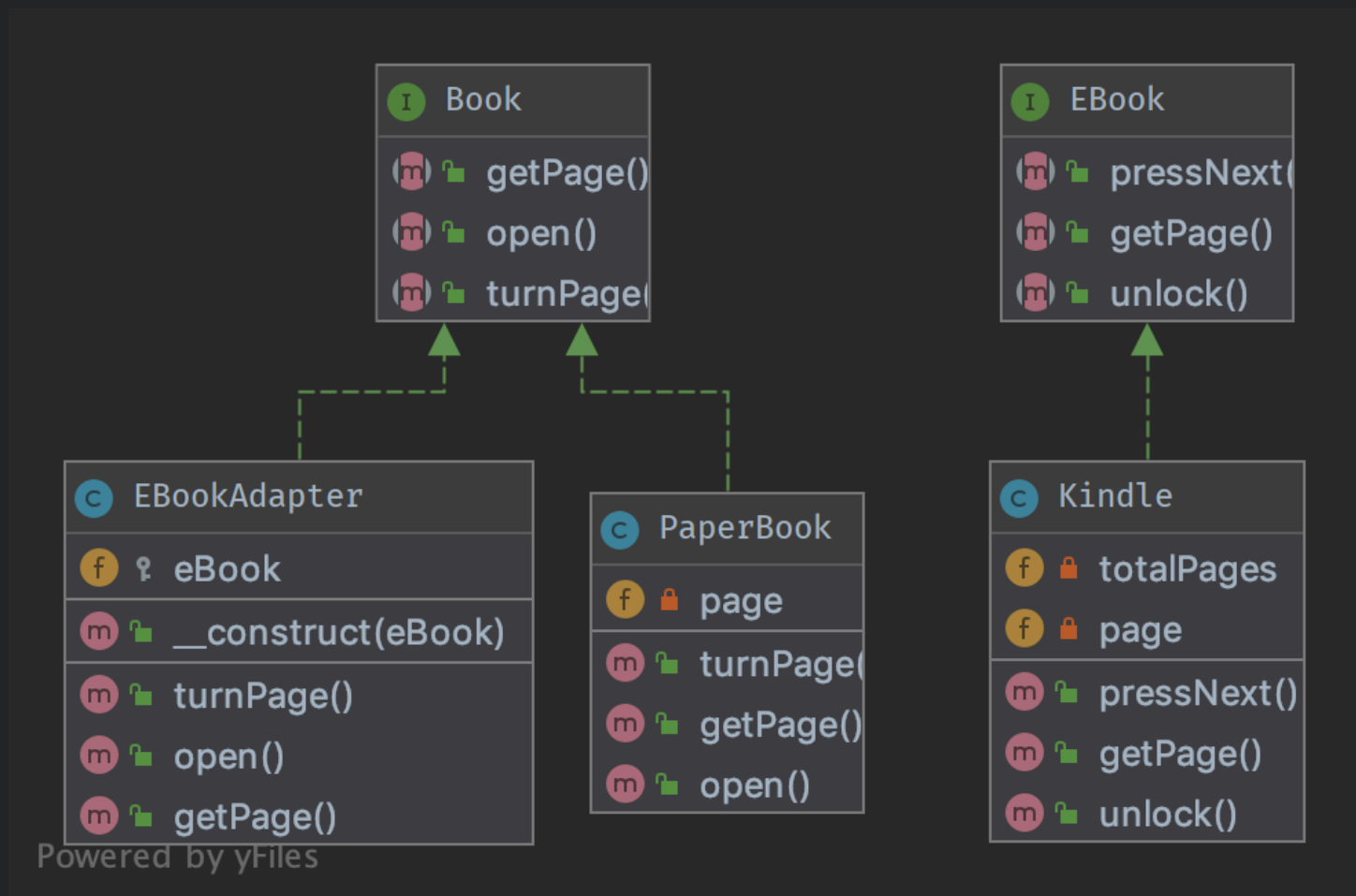


**Адаптер** - это структурный паттерн, который позволяет объектам с несовместимыми интерфейсами работать вместе.

Примеры:

- адаптер клиентских библиотек для работы с базами данных
- приведение интерфейсов различных api сервисов к одному виду



## Реализация:

```
<?php

interface Book
{
    public function turnPage();
    public function open();
    public function getPage(): int;
}

class PaperBook implements Book
{
    private int $page;
    public function open(): void
    {
        $this->page = 1;
    }
    public function turnPage(): void
    {
        $this->page++;
    }
    public function getPage(): int
    {
        return $this->page;
    }
}

interface EBook
{
    public function unlock();
    public function pressNext();
    public function getPage(): array;
}

class BookAdapter implements Book
{
    protected EBook $ebook;
    public function __construct(EBook $ebook)
    {
        $this->ebook = $ebook;
    }
    public function open(): void
    {
        $this->ebook->unlock();
    }
    public function turnPage(): void
    {
        $this->ebook->pressNext();
    }
    public function getPage(): int
    {
        return $this->ebook->getPage()[0];
    }
}
```

```
class ReadBook implements EBook
{
    private int $page = 1;
    private int $totalPages = 100;
    public function pressNext(): void
    {
        $this->page++;
    }
    public function unlock()
    {
    }
    public function getPage(): array
    {
        return [$this->page, $this->totalPages];
    }
}

$book = new PaperBook();
$book->open();
$book->turnPage();

$readBook = new ReadBook();
$book = new BookAdapter($readBook);
$book->open();
$book->turnPage();
```