

Typescript

// Абстрактный класс Publisher

```
abstract class Publisher {
```

```
    title: string;
```

```
    author: string;
```

```
    pubYear: number;
```

```
    copies: number;
```

```
    constructor(title: string, author: string, pubYear: number, copies: number) {
```

```
        this.title = title;
```

```
        this.author = author;
```

```
        this.pubYear = pubYear;
```

```
        this.copies = copies;
```

```
    }
```

```
    getTitle(): string {
```

```
        return this.title;
```

```
    }
```

```
    setTitle(newTitle: string): void {
```

```
        this.title = newTitle;
```

```
    }
```

```
    // Добавьте методы для получения и изменения других полей
```

```
}
```

// Подклассы Book и Magazine

```
class Book extends Publisher {
```

```
    pages: number;
```

```
    constructor(title: string, author: string, pubYear: number, copies: number, pages: number) {
```

```
        super(title, author, pubYear, copies);
```

```
        this.pages = pages;
```

```
    }
```

```
}
```

```
class Magazine extends Publisher {
```

```
    issue: number;
```

```
    constructor(title: string, author: string, pubYear: number, copies: number, issue: number) {
```

```
        super(title, author, pubYear, copies);
```

```
        this.issue = issue;
```

```
    }
```

```
}
```

// Интерфейс Reception

```
interface Reception {
```

```
    delivery(): void;
```

```

    receive(): void;
}

// Реализация интерфейса для Book и Magazine
class Reader {
    firstName: string;
    lastName: string;
    items: Publisher[] = [];

    constructor(firstName: string, lastName: string) {
        this.firstName = firstName;
        this.lastName = lastName;
    }

    // Методы выдачи и принятия издания у читателя
    borrowItem(item: Publisher): void {
        if (this.items.length < 3 && item.copies > 0) {
            this.items.push(item);
            item.copies--;
            console.log(`${this.firstName} ${this.lastName} взял ${item.title}`);
        } else {
            console.log(`Невозможно выдать ${item.title} для ${this.firstName} ${this.lastName}`);
        }
    }

    returnItem(item: Publisher): void {
        const index = this.items.indexOf(item);
        if (index !== -1) {
            this.items.splice(index, 1);
            item.copies++;
            console.log(`${this.firstName} ${this.lastName} вернул ${item.title}`);
        }
    }
}

// Класс Library
class Library {
    publications: Publisher[] = [];

    addPublication(publication: Publisher): void {
        this.publications.push(publication);
    }

    removePublication(publication: Publisher): void {
        const index = this.publications.indexOf(publication);
        if (index !== -1) {
            this.publications.splice(index, 1);
        }
    }
}

```

```
    }  
  }  
}
```

```
// Тестирование методов
```

```
const book1 = new Book("Приключения", "Автор1", 2021, 5, 200);  
const magazine1 = new Magazine("Журнал", "Автор2", 2022, 3, 10);
```

```
const reader1 = new Reader("Иван", "Иванов");  
reader1.borrowItem(book1);  
reader1.borrowItem(magazine1);  
reader1.borrowItem(book1);
```

```
const library = new Library();  
library.addPublication(book1);  
library.addPublication(magazine1);
```

```
console.log(reader1.items.map(item => item.title));  
console.log(book1.copies);  
console.log(magazine1.copies);
```