

Komanda pradžioje išsirenka komandos lyderį, kuris sukurs projekto aplanką, įkels į savo GitHub paskyrą, kiti komandos nariai „klonuos“ repozitoriją. Komandos lyderis prideda komandos narius (collaborators) prie repozitorijos.

Apsitariat dėl projekto architektūros, dėl tvarkos ir stiliaus kaip bus vadinamos klasės, metodai, kintamieji, bendra projekto tvarka turi išlikti vienoda, nežiūrinti to kiek dirba programuotojų.

Galimas projekto išsiskirstymas į atskiras užduotis:

- > Projekto architektūros apsibrėžimas,
- > Klasės Book apsirašymas,
- > Klasės Category apsirašymas,
- > Klasės Library apsirašymas,
- > Klasės Reader apsirašymas,
- > Klasės Rating apsirašymas,
- > Bazinių šablono susidėliojimas (HTML + CSS),
- > Visų dienos darbų apjungimas ir visos komandos repozitorijų susivienodinimas (push/pull),
- > Atlikto darbo išsistestavimas,

Apmastykite projektą, galima papildyti funkcionalumu. Pridėti papildomų klasių, savybių ar metodų klasėms jei manot, kad reikia. Pabaigę išsistestuokite klases naršyklės konsolėje, ar visi metodai taip veikia kaip tikėtasi. Dirbkite komandiškai, svarbiausia, kad kodo nepersiuntinėtumėte komentaruose 😊, o pushintumėte ir pull'intumėte iš /į repozitoriją.

P.S. Dėl unikalaus klasės objekto id - klasė turi statinį kintamąjį counter, kuris skaičiuoja sukurtus klasės egzempliorius (tai ir galima priskirti objekto id).

# LIBRARY

## 1. Book Klasė

Aprašymas: Klasė, kuri reprezentuoja knygą.

### Savybės:

id: unikalus knygos numeris

title: knygos pavadinimas.

author: knygos autorius.

isbn: knygos ISBN numeris.

price: knygos kaina.

description: trumpas knygos aprašymas.

is\_checked\_out: rodoma, ar knyga yra pasiskolinta.

### Metodai:

Geteriai, seteriai

checkAvailability(): patikrina, ar knyga yra pasiskolinta.

checkOut(): pažymi knygą kaip pasiskolintą.

checkIn(): pažymi knygą kaip prieinamą.

## 2. Category Klasė

Aprašymas: Klasė, kuri reprezentuoja knygų kategoriją.

### Savybės:

id: unikalus kategorijos numeris

categoryName: kategorijos pavadinimas.

books: masyvas, kuriame saugomos knygos.

### Metodai:

Geteriai, seteriai

getBooksByAuthor(authorName): grąžina masyvą knygų, kurių autorius atitinka duotą vardą.

getBooksByPriceRange(minPrice, maxPrice): grąžina knygas, kurių kaina yra nustatytame intervale.

### **3. Library Klasė**

Aprašymas: Klasė, kuri saugo visas kategorijas ir yra atsakinga už bibliotekos valdymą.

#### **Savybės:**

categories: masyvas, kuriame saugomos kategorijos objektai.

#### **Metodai:**

Geteriai, seteriai

### **4. Reader (Skaitytojas) Klasė**

Aprašymas: Klasė, kuri reprezentuoja bibliotekos skaitytoją.

#### **Savybės:**

name: skaitytojo vardas.

reader\_id: unikalus skaitytojo identifikatorius.

borrowed\_books: masyvas, kuriame saugomos pasiskolintos knygos.

#### **Metodai:**

getName(): grąžina skaitytojo vardą.

setName(name): nustato skaitytojo vardą.

getReaderId(): grąžina skaitytojo ID.

setReaderId(reader\_id): nustato skaitytojo ID.

borrowBook(book): prideda knygą į pasiskolintų knygų sąrašą, jei ji yra prieinama.

returnBook(book): pašalina knygą iš pasiskolintų knygų sąrašo.

getBorrowedBooks(): grąžina visų pasiskolintų knygų masyvą.

### **5. Rating Klasė (Klientų atsiliepimų sistema (Feedback))**

Aprašymas: Klasė, leidžianti skaitytojams palikti atsiliepimus apie knygas.

#### **Savybės:**

id: unikalus atsiliepimo numeris.

bookId: nuoroda į knygos id.

readerId: nuoroda į skaitytojo id.

rating: įvertinimas (1-5).

comment: atsiliepimo tekstas.

#### **Metodai:**

Geteriai, seteriai