
PROJEKT 1

Zarządzanie biblioteką

1. Stwórz następującą strukturę klas:

- User
 - Student
 - Teacher
- Resource – klasa abstrakcyjna
 - Book
 - Paperback
 - ScientificBook
 - Audiobook
 - EBook
 - Journal
 - ScientificJournal
- Interfejs – Digital
Implementowany przez Audiobook i EBook
- Intefejs – Rentable
Implementowany przez Paperback i Journal
- Interfejs – Scientific
Implementowany przez ScientificBook and ScientificJournal

2. Do klas dodaj następujące pola. Można dodać też inne, ale te wypisane tutaj są obowiązkowe:

- User: name, surname, idUser, department
- Student: course
- Teacher: position
- Resource: title, publisher, resourceId
- Book: author
- Paperback: pages, rentDate, dueDate, user
- ScientificBook: domain
- Audiobook: length, numberOfDownload
- E-book: size, numberOfDownload
- Journal: publishDate, rentDate, dueDate, user
- ScientificJournal: domain

UWAGA! W klasach mogą znajdować się również inne pola i metody. Również w projekcie mogą znajdować się dodatkowe klasy.

3. Dodaj metody w interfejsach i zaimplementuj je w odpowiednich miejscach:

- Rentable: rent(user) – data wypożyczenia to aktualna data, długość wypożyczenia dla studenta to 1 miesiąc, dla nauczyciela 3 miesiące dla książek i 10 dni dla czasopism; oraz metodę getUser()
- Digital: download() – zlicza ilość pobrań

- Scientific: getDomain()
4. Dodaj metody w klasie Resource:
 - wypisującą informacje o zasobie (bez informacji o wypożyczeniach czy pobraniach) i nadpisz ją w klasach dziedziczących.
 5. Dodaj typ wliczeniowy Status z wartościami: AVAILABLE, UNAVAILABLE, DIGITAL
 6. Dodaj do Resource metodę getStatus, która zwraca status.
 7. Dodaj nowy wyjątek NoAvailableResource, który ma być rzucany przy próbie wypożyczenia zasobu już wypożyczonej.
 8. Dodaj w klasie User następujące metody:
 - numberOfResources() – zwracającą ilość wypożyczonych zasobów
 - getFee() – jeśli zasób nie został oddany o czasie naliczana jest kara 1zł – 1 dzień
 9. Dodaj metodę w klasie Book:
 - isTheSameBook(Book book1) – sprawdzająca czy to są egzemplarze tej samej książki (ten sam tytuł i autor)
 10. Dodaj metodę w interfejsie Scientific
 - isTheSameDomain(Scientific res2)
 11. Skompiluj z klasą Main (z klasy nie można nic usuwać, ale można dodawać).
 12. W klasie Main zaimplementuj metodę printAllPaperBooks. Metoda ta ma wypisać wszystkie różne książki oraz informacje ile jest egzemplarzy danej książki i ile z nich jest wypożyczonych.
 13. Przetestuj metody numberOfResources i getFee z klasy User.
 14. Przetestuj metody isTheSameBook i isTheSameDomain.
 15. Przetestuj metodę rent z interfejsu Rentable (na dowolnej z klas np. PaperBook).

METODA MAIN:

```
public static void main(String[] args) {
    User[] users = new User[5];
    users[0]=new Student("Jan","Kowalski",1,"Matematyki i Informatyki","matematyka");
    users[1]=new Student("Anna", "Nowak", 2, "Matematyki i Informatyki", "informatyka");
    users[2]=new Student("Alicja", "Lis", 3, "Prawa i Administracji", "prawo");
    users[3]=new Teacher("Aleksandra", "Zakrzewska", 4, "Matematyki i Informatyki", "adiunkt");
    users[4]=new Teacher("Karl", "Przybylski", 5, "Prawa i Administracji", "profesor");

    Resource[] resources = new Resource[19];
    resources[0]=new PaperBook("Pan Tadeusz","GREG",1,"Adam Mickiewicz",334);
    resources[1]=new PaperBook("Pan Tadeusz","GREG",2,"Adam Mickiewicz",334);
    resources[2]=new PaperBook("Pan Tadeusz","GREG",3,"Adam Mickiewicz",334);
    resources[3]=new ScientificBook("Thinking in Java", "Helion",4, "Bruce Eckel", 1248,
    "informatyka");
    resources[4]=new ScientificBook("Prawo karne - część ogólna", "Wolters Kluwer", 5, "Jarosław
    Warylewski",680,"prawo");
    resources[5]=new ScientificBook("Prawo karne - część ogólna", "Wolters Kluwer", 6, "Jarosław
    Warylewski",680,"prawo");
    resources[6]=new PaperBook("Hrabia Monte Cristo - I", "Świat Książki",7, "Aleksander Dumas",
    672);
    resources[7]=new PaperBook("Hrabia Monte Cristo - I", "Świat Książki",8, "Aleksander Dumas",
    672);
    resources[8]=new Audiobook("Pan Tadeusz", "Aleksandria", 9, "Adam Mickiewicz", 470);
    resources[9]=new Audiobook("Hobbit", "Muza", 10, "J. R. R. Tolkien",636);
```

```

resources[10]=new EBook("Hobbit", "Iskry", 11, "J. R. R. Tolkien", 3.1);
resources[11]=new EBook("Lałka", "Agora", 12, "Bolesław Prus", 3.4);
resources[12]=new Journal("Książki", "wyborcza", 13, LocalDate.of(2024, 9, 01));
resources[13]=new Journal("Książki", "wyborcza", 14, LocalDate.of(2024, 9, 01));
resources[14]=new Journal("Książki", "wyborcza", 15, LocalDate.of(2024, 9, 01));
resources[15]=new Journal("Książki", "wyborcza", 16, LocalDate.of(2024, 9, 01));
resources[16]=new Journal("Książki", "wyborcza", 17, LocalDate.of(2024, 9, 01));
resources[17]=new ScientificJorunal("Państwo i Prawo", "Wolters
Kluwer", 18, LocalDate.of(2024, 8, 1), "prawo");
resources[18]=new ScientificJorunal("Państwo i Prawo", "Wolters
Kluwer", 18, LocalDate.of(2024, 8, 1), "prawo");

PaperBook book1=(PaperBook)resources[0];
book1.print();
try {
    book1.rent(users[0]); //wypożyczenie bez problemu
} catch (NoAvailableResourceException e) {
    System.err.println("Book unavailable");
}

System.out.println(book1.getUser().name+" "+book1.getUser().surname+" due date:
"+book1.dueDate);
System.out.println(book1.getStatus()); //Status UNAVAILABLE

try {
    book1.rent(users[1]); //Rzucenie wyjątku
} catch (NoAvailableResourceException e) {
    System.err.println("Book unavailable");
}

EBook book2=(EBook)resources[10];
book2.print();
book2.download();
book2.download();
System.out.println("dowlands: "+book2.numberOfDownloads); //2 wypożyczenia
System.out.println(book2.getStatus()); //Status DIGITAL

ScientificJorunal journal1=(ScientificJorunal)resources[17];
journal1.print();
System.out.println(journal1.getDomain()); //prawo
System.out.println(journal1.getStatus()); //status AVAILABLE
try {
    journal1.rent(users[1]); //wypożyczenie bez problemu
} catch (NoAvailableResourceException e) {
    throw new RuntimeException(e);
}

PaperBook book3=(PaperBook)resources[2];

```

```

book3.print();
System.out.println(book3.getStatus()); //status AVAILABLE

ScientificBook book4=(ScientificBook)resources[4];
book4.print();
try {
    book4.rent(users[0]); //wypożyczenie bez problemu
} catch (NoAvailableResourceException e) {
    System.err.println("Book unavailable");
}

System.out.println("User 0 "+users[0].numberOfResources(resources)); //2 zasoby
System.out.println("User 1 "+users[1].numberOfResources(resources)); //1 zasób
System.out.println("User 2 "+users[2].numberOfResources(resources)); //0 zasobów

book1.dueDate=LocalDate.now().minusDays(5);
System.out.println("User 0 fee:"+users[0].getFee(resources)); //5 zł kary

System.out.println(book1.isTheSameBook(book4)); //fałsz
System.out.println(book1.isTheSameBook(book3)); //prawda

Audiobook book5 = (Audiobook)resources[8];
System.out.println(book1.isTheSameBook(book5)); //prawda

System.out.println(book4.isTheSameDomain(journal1)); //prawda

ScientificBook book6 = (ScientificBook) resources[3];
System.out.println(book6.isTheSameDomain(journal1)); //fałsz

printAllPaperBooks(resources);
//Pan Tadeusz Adam Mickiewicz available: 2 unavailable: 1
//Thinking in Java Bruce Eckel available: 1 unavailable: 0
//Prawo karne - część ogólna Jarosław Warylewski available: 1 unavailable: 1
//Hrabia Monte Cristo - I Aleksander Dumas available: 2 unavailable: 0
}

```