

Uppgift 1: Bibliotekssystem (5 personer)

Beskrivning

Skapa ett bibliotekssystem där olika gruppmedlemmar ansvarar för olika moduler. Fokus på grundfunktionalitet som fungerar tillsammans.

Gruppindelning

Person 1 - Bokhantering (2-3 funktioner):

```
import java.util.ArrayList;

public static void addBook(ArrayList<String> titles,
    ArrayList<String> authors, ArrayList<String> isbn, String
    title, String author, String isbnNumber)

public static void displayAllBooks(ArrayList<String> titles,
    ArrayList<String> authors, ArrayList<String> isbn)

public static int searchBook(ArrayList<String> titles,
    ArrayList<String> authors, String searchTerm)
```

Person 2 - Lånesystem (3 funktioner):

```
public static boolean borrowBook(ArrayList<Boolean> available,
    ArrayList<String> borrowers, ArrayList<String>
    borrowedBooks, int bookIndex, String borrowerName)

public static boolean returnBook(ArrayList<Boolean> available,
    ArrayList<String> borrowers, ArrayList<String>
    borrowedBooks, String isbnNumber)

public static void displayBorrowedBooks(ArrayList<String>
    borrowers, ArrayList<String> borrowedBooks)
```

Person 3 - Användarhantering (3 funktioner):

```
public static void registerUser(ArrayList<String> userNames,
    ArrayList<String> phoneNumbers, String name, String
    phoneNumber)
```

```
public static void displayAllUsers(ArrayList<String> userNames,  
    ArrayList<String> phoneNumbers)
```

```
public static int searchUser(ArrayList<String> userNames, String  
    name)
```

Person 4 - Enkla Rapporter (3 funktioner):

```
public static int countAvailableBooks(ArrayList<Boolean>  
    available)
```

```
public static int countBorrowedBooks(ArrayList<Boolean>  
    available)
```

```
public static void displayLibraryStatistics(ArrayList<String>  
    titles, ArrayList<Boolean> available, ArrayList<String>  
    userNames)
```

Person 5 - Meny och Integration (2-3 funktioner):

```
public static void displayMainMenu()
```

```
public static void displayBookMenu()
```

```
public static void displayLoanMenu()
```

```
// + ansvarar för main-metoden och att koppla ihop alla delar
```

Förenklad datastruktur

```
// I main-metoden - ENDAST dessa listor för att spara tid
```

```
// Böcker (index motsvarar varandra)
```

```
ArrayList<String> bookTitles = new ArrayList<>();
```

```
ArrayList<String> bookAuthors = new ArrayList<>();
```

```
ArrayList<String> bookISBN = new ArrayList<>();
```

```
ArrayList<Boolean> bookAvailable = new ArrayList<>(); // true =  
    tillgänglig
```

```
// Lån (index motsvarar varandra)
```

```
ArrayList<String> borrowerNames = new ArrayList<>();
```

```
ArrayList<String> borrowedBooks = new ArrayList<>(); // ISBN för  
    lånad bok
```

```
// Användare (index motsvarar varandra)
```

```
ArrayList<String> userNames = new ArrayList<>();  
ArrayList<String> phoneNumbers = new ArrayList<>();
```

Förinställd testdata (Person 5 skapar detta)

```
// Lägg till dessa i början av main för att spara tid på  
    inmatning
```

```
// Fördefinierade böcker  
bookTitles.add("Harry Potter");  
bookTitles.add("Sagan om ringen");  
bookTitles.add("1984");  
bookAuthors.add("J.K. Rowling");  
bookAuthors.add("Tolkien");  
bookAuthors.add("Orwell");  
bookISBN.add("111");  
bookISBN.add("222");  
bookISBN.add("333");  
bookAvailable.add(true);  
bookAvailable.add(true);  
bookAvailable.add(false); // 1984 är utlånad
```

```
// Fördefinierade användare  
userNames.add("Anna");  
userNames.add("Erik");  
phoneNumbers.add("070-1234567");  
phoneNumbers.add("070-7654321");
```

```
// Fördefinierat lån  
borrowerNames.add("Anna");  
borrowedBooks.add("333"); // Anna har lånat 1984
```

Enkel huvudmeny (Person 5)

```
public static void main(String[] args) {  
    Scanner scanner = new Scanner(System.in);  
  
    // Initiera listor och testdata här...  
  
    while (true) {  
        System.out.println("\n=== BIBLIOTEKSSYSTEM ===");  
        System.out.println("1. Visa alla böcker");  
        System.out.println("2. Lägg till bok");  
        System.out.println("3. Låna bok");
```

```

System.out.println("4. Återlämna bok");
System.out.println("5. Visa statistik");
System.out.println("0. Avsluta");

int choice = scanner.nextInt();
scanner.nextLine(); // Rensa newline

switch (choice) {
    case 1:
        displayAllBooks(bookTitles, bookAuthors,
            bookISBN);
        break;
    case 2:
        System.out.print("Titel: ");
        String title = scanner.nextLine();
        System.out.print("Författare: ");
        String author = scanner.nextLine();
        System.out.print("ISBN: ");
        String isbn = scanner.nextLine();
        addBook(bookTitles, bookAuthors, bookISBN, title,
            author, isbn);
        break;
    // Fortsätt för andra val...
}
}
}

```

Exempel på färdig funktion (för referens)

```

public static void addBook(ArrayList<String> titles,
    ArrayList<String> authors, ArrayList<String> isbn, String
    title, String author, String isbnNumber) {
    titles.add(title);
    authors.add(author);
    isbn.add(isbnNumber);
    bookAvailable.add(true); // Ny bok är alltid tillgänglig
    System.out.println("Bok tillagd: " + title + " av " +
        author);
}

```

Tips för att lyckas på kort tid:

1. **Håll det enkelt** - Inga komplicerade algoritmer
2. **Använd testdata** - Slipp skriva in böcker manuellt
3. **Testa löpande** - Kör koden ofta för att hitta fel tidigt

4. **Kommunicera** - Fråga varandra om ni fastnar
5. **Fokus på integration** - Bättre med enklare funktioner som fungerar tillsammans

Minimum viable product (MVP):

- Kan lägga till böcker ✓
- Kan visa alla böcker ✓
- Kan låna en bok ✓
- Kan återlämna en bok ✓
- Fungerande meny ✓

Om ni har tid över:

- Förbättra sökfunktionen
- Lägg till fler rapporter
- Validering av ISBN-format
- Bättre felhantering