

Project Explanation: Generic Library Catalog

This project, **Generic Library Catalog**, is a versatile Java application that uses generics to manage a flexible library catalog for different types of library items (e.g., books, DVDs, magazines). It demonstrates the application of generic classes and methods in a real-world context. The project consists of the following key components:

1. **LibraryItem Class:**

- Represents individual library items with attributes such as `title`, `author`, and `itemID`.
- Provides a reusable structure for various types of library items.

2. **GenericCatalog Class:**

- Implements a generic catalog that can store and manage different types of items.
- Includes methods for adding, removing, and displaying items in the catalog.
- Uses generics to ensure flexibility and compatibility with multiple item types.

3. **LibraryCatalogApp Class (Main Class):**

- Provides a simple command-line interface for users to interact with the library catalog.
- Enables users to add new library items, remove existing ones, and view the catalog.
- Manages the program flow and user interaction.

The project showcases:

- **Generic Programming Concepts:** Demonstrates the use of generic classes and methods for flexibility and code reusability.
- **Error Handling:** Handles errors gracefully, such as when attempting to remove a non-existent item, with clear and user-friendly messages.
- **Command-Line Interface:** Allows users to manage the library catalog with straightforward input and output.
- **Modularity:** Separates concerns into different classes to improve code organization and readability.

Instructions to Run and Test the Program

1. **Extract the Folder:**

- Unzip the folder named `Generic Library Catalog` and navigate to its directory.

2. **Compile and Run:**

Use the following commands to compile and run the program:

To compile:

```
javac -cp src src/com/uopeople/amm/LibraryCatalogApp.java -d bin
```

To run:

```
java -cp bin com.uopeople.amm.LibraryCatalogApp
```

- On Windows, ensure the paths use backslashes (`\`) if required.

1. Observe Output:

- The console will display a menu with options to add items, remove items, and view the catalog.
- Example interaction: Library Catalog Menu: 1. Add Item 2. Remove Item 3. Display Catalog 4. Exit Choose an option: 1 Enter Title: Java Programming Enter Author: John Doe Enter Item ID: BK001 Item added: LibraryItem{title='Java Programming', author='John Doe', itemID='BK001'}

2. To Test:

- Add different types of library items, such as books, DVDs, or magazines, by modifying the LibraryItem structure.
- Test scenarios where a non-existent item is removed to observe error handling.
- Validate the generic catalog's compatibility by attempting to store items of varying types.

3. Terminate Execution:

- Exit the program through the menu by selecting the Exit option.
- The program handles all cleanup and exits gracefully.

Example Screenshots:

```
java -cp bin com.uopeople.amm.LibraryCatalogApp
```

```
Library Catalog Menu:
```

```
1. Add Item  
2. Remove Item  
3. Display Catalog  
4. Exit
```

```
Choose an option: █
```

```
Library Catalog Menu:
```

```
1. Add Item  
2. Remove Item  
3. Display Catalog  
4. Exit
```

```
Choose an option: 1
```

```
Enter Title: The art of not giving a fuck
```

```
Enter Author: whoever writes it
```

```
Enter Item ID: 1
```

```
Item added: LibraryItem{title='The art of not giving a fuck', author='whoever writes it', itemID='1'}
```

```
Library Catalog Menu:
1. Add Item
2. Remove Item
3. Display Catalog
4. Exit
Choose an option: 3
Catalog Items:
LibraryItem{title='The art of not giving a fuck', author='whoever writes it', itemID='1'}
```

```
Library Catalog Menu:
1. Add Item
2. Remove Item
3. Display Catalog
4. Exit
Choose an option: 2
Enter Item ID to remove: 1
Item removed: LibraryItem{title='The art of not giving a fuck', author='whoever writes it', itemID='1'}

Library Catalog Menu:
1. Add Item
2. Remove Item
3. Display Catalog
4. Exit
Choose an option: 3
Catalog Items:
```