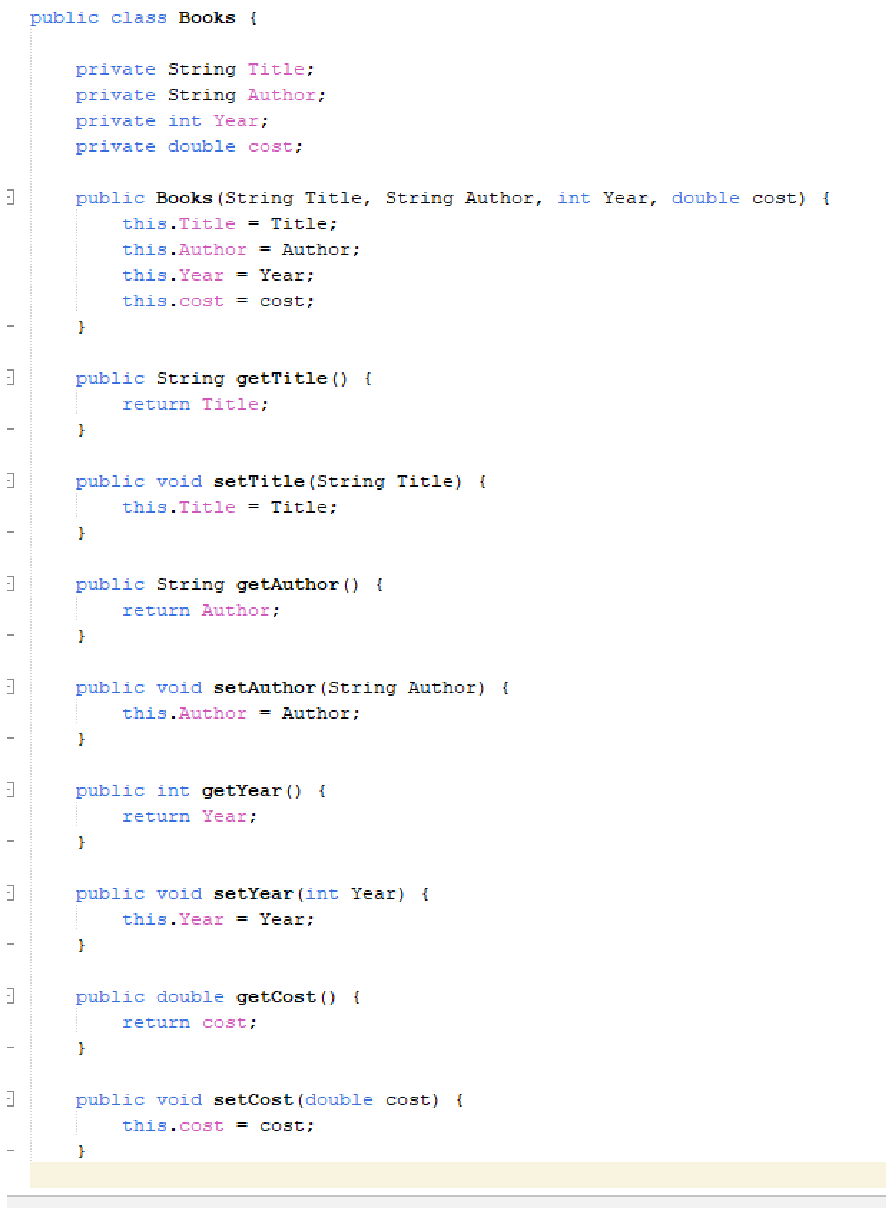
Streams

Regina Rodriguez

- **Stream 1**

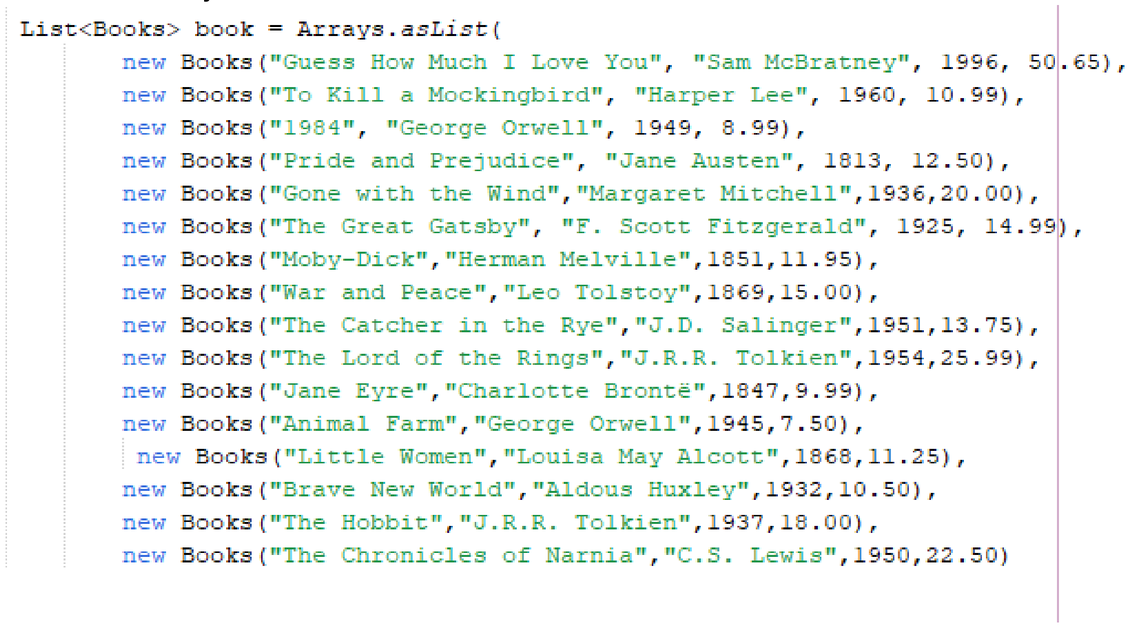
  Our problem involves books that have their titles, authors, and years of publication, and we want to display the titles of the books published after 1920.

  First we created our Class “Books” with its attributes and constructors.



```
public class Books {

    private String Title;
    private String Author;
    private int Year;
    private double cost;

    public Books(String Title, String Author, int Year, double cost) {
        this.Title = Title;
        this.Author = Author;
        this.Year = Year;
        this.cost = cost;
    }

    public String getTitle() {
        return Title;
    }

    public void setTitle(String Title) {
        this.Title = Title;
    }

    public String getAuthor() {
        return Author;
    }

    public void setAuthor(String Author) {
        this.Author = Author;
    }

    public int getYear() {
        return Year;
    }

    public void setYear(int Year) {
        this.Year = Year;
    }

    public double getCost() {
        return cost;
    }

    public void setCost(double cost) {
        this.cost = cost;
    }
```

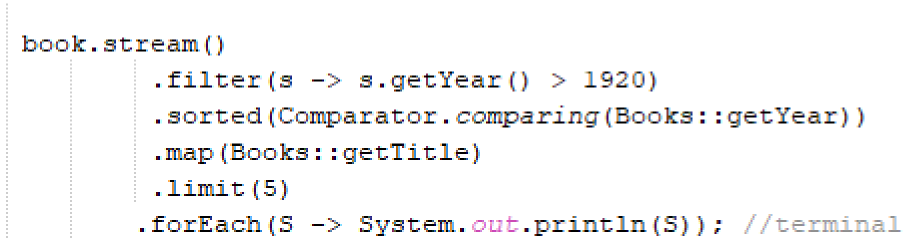
Creates our objects



```
List<Books> book = Arrays.asList(
        new Books("Guess How Much I Love You", "Sam McBratney", 1996, 50.65),
        new Books("To Kill a Mockingbird", "Harper Lee", 1960, 10.99),
        new Books("1984", "George Orwell", 1949, 8.99),
        new Books("Pride and Prejudice", "Jane Austen", 1813, 12.50),
        new Books("Gone with the Wind","Margaret Mitchell",1936,20.00),
        new Books("The Great Gatsby", "F. Scott Fitzgerald", 1925, 14.99),
        new Books("Moby-Dick","Herman Melville",1851,11.95),
        new Books("War and Peace","Leo Tolstoy",1869,15.00),
        new Books("The Catcher in the Rye","J.D. Salinger",1951,13.75),
        new Books("The Lord of the Rings","J.R.R. Tolkien",1954,25.99),
        new Books("Jane Eyre","Charlotte Brontë",1847,9.99),
        new Books("Animal Farm","George Orwell",1945,7.50),
         new Books("Little Women","Louisa May Alcott",1868,11.25),
        new Books("Brave New World","Aldous Huxley",1932,10.50),
        new Books("The Hobbit","J.R.R. Tolkien",1937,18.00),
        new Books("The Chronicles of Narnia","C.S. Lewis",1950,22.50)
```

This are the steps that we have to follow to complete our problem

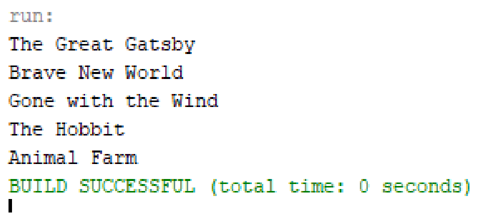
1. Filter the books published after 1920.
2. Sort the books by their year of publication in ascending order.
3. Map the results to include only the titles of each book.
4. Limit the results to the first 5 book titles.



```
book.stream()
        .filter(s -> s.getYear() > 1920)
        .sorted(Comparator.comparing(Books::getYear))
        .map(Books::getTitle)
        .limit(5)
       .forEach(S -> System.out.println(S)); //terminal
```

We create a stream of our books and use the filter method to ensure only books with publication years after 1920 are processed. Then we apply sorted to order the books from the oldest to the newest publication year. Next, we use map to extract only the book titles. Finally, we limit the results to the first 5 titles and use forEach to print each title.

Result:



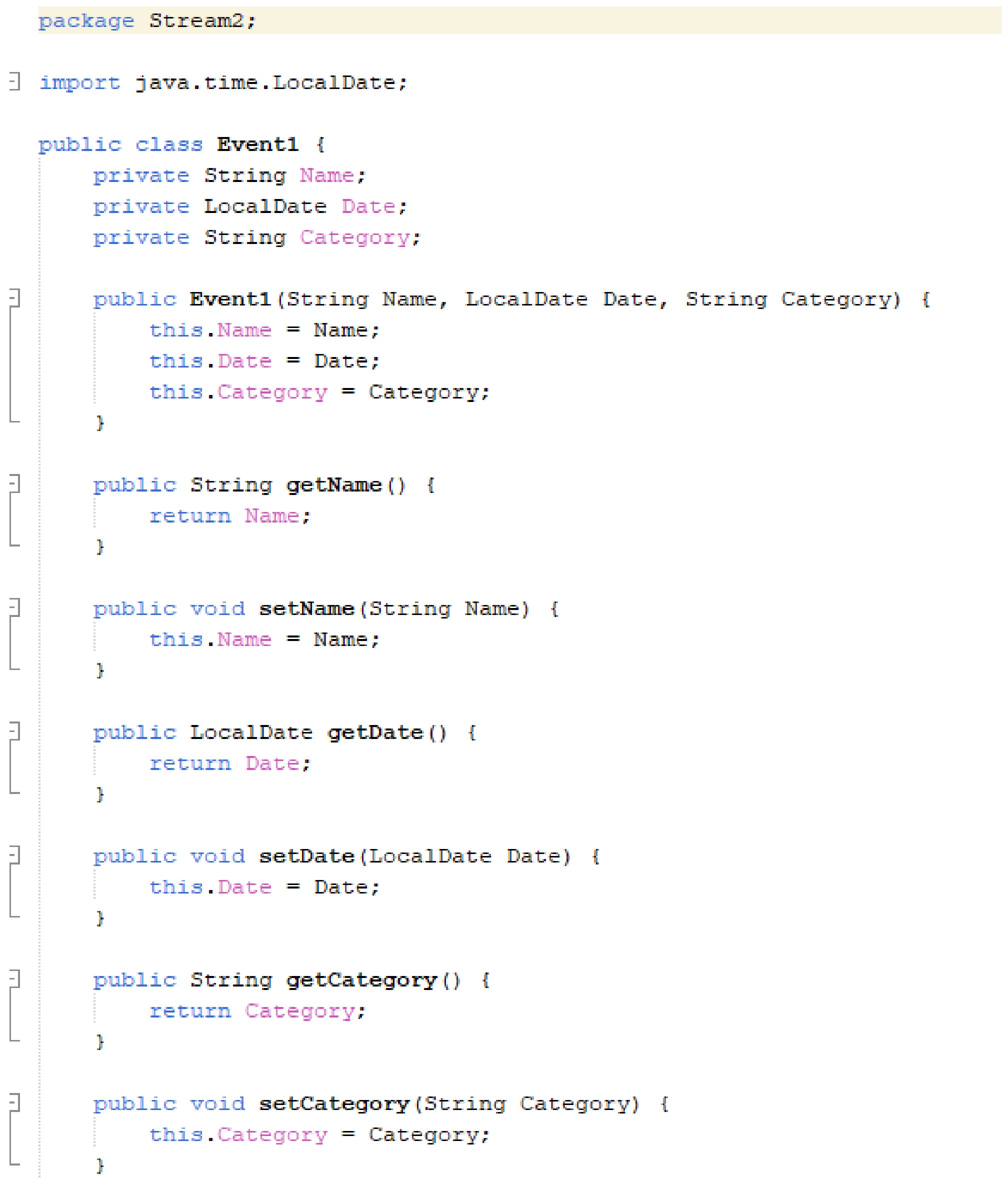
```
run:
The Great Gatsby
Brave New World
Gone with the Wind
The Hobbit
Animal Farm
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Stream 2**

Our problem involves managing a party business, and we need to identify events that will occur after a certain date and ensure that no event is repeated. We want to achieve this through four steps:

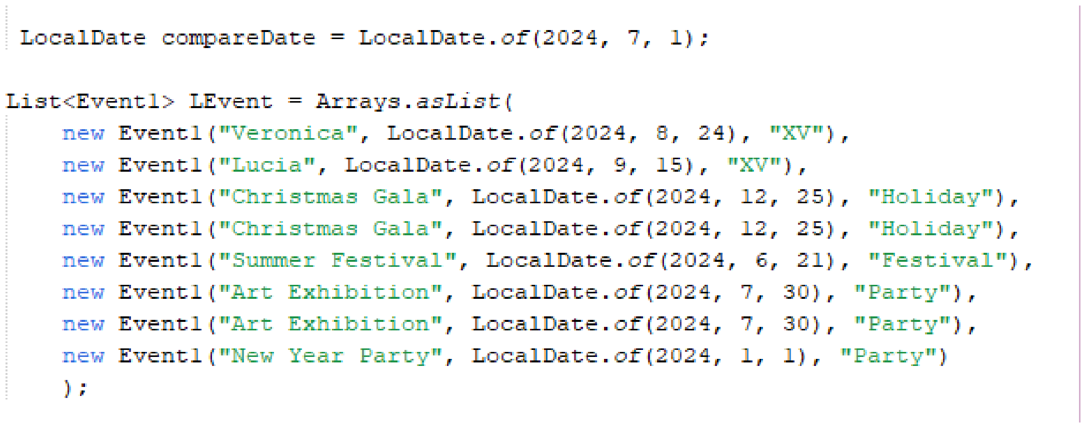
1. Sort the events by category.
2. Filter to only include events that occur after 2024/07/01.
3. Map the results to include only the category of each event.
4. Remove any duplicate events.

But first, we need to create our classes Events.



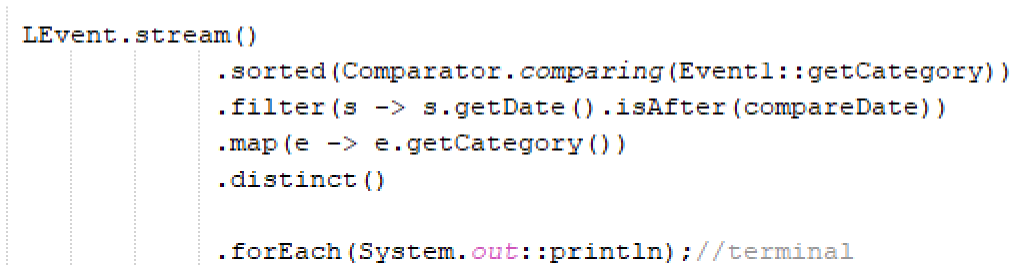
```java
package Stream2;

import java.time.LocalDate;

public class Event1 {
    private String Name;
    private LocalDate Date;
    private String Category;

    public Event1(String Name, LocalDate Date, String Category) {
        this.Name = Name;
        this.Date = Date;
        this.Category = Category;
    }

    public String getName() {
        return Name;
    }

    public void setName(String Name) {
        this.Name = Name;
    }

    public LocalDate getDate() {
        return Date;
    }

    public void setDate(LocalDate Date) {
        this.Date = Date;
    }

    public String getCategory() {
        return Category;
    }

    public void setCategory(String Category) {
        this.Category = Category;
    }
```

Creates our objects



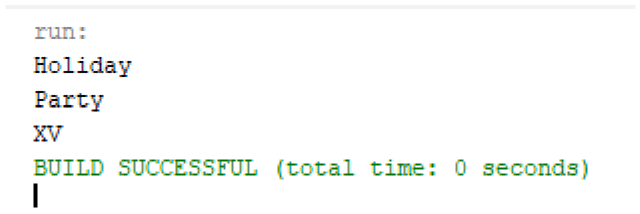
```
 LocalDate compareDate = LocalDate.of(2024, 7, 1);

List<Event1> LEvent = Arrays.asList(
    new Event1("Veronica", LocalDate.of(2024, 8, 24), "XV"),
    new Event1("Lucia", LocalDate.of(2024, 9, 15), "XV"),
    new Event1("Christmas Gala", LocalDate.of(2024, 12, 25), "Holiday"),
    new Event1("Christmas Gala", LocalDate.of(2024, 12, 25), "Holiday"),
    new Event1("Summer Festival", LocalDate.of(2024, 6, 21), "Festival"),
    new Event1("Art Exhibition", LocalDate.of(2024, 7, 30), "Party"),
    new Event1("Art Exhibition", LocalDate.of(2024, 7, 30), "Party"),
    new Event1("New Year Party", LocalDate.of(2024, 1, 1), "Party")
    );
```

We create a stream of events, but first, we need to set up an attribute to limit the date we're interested in. After that, we use sorted to order the events by category in alphabetical order using ASCII values. Then we filter the events based on the date. Next, we use map to extract only the event categories, and finally, we remove any duplicate events. We use forEach to print out and view the most recent event categories.



```
LEvent.stream()
                .sorted(Comparator.comparing(Event1::getCategory))
                .filter(s -> s.getDate().isAfter(compareDate))
                .map(e -> e.getCategory())
                .distinct()

                .forEach(System.out::println);//terminal
```

Result:



```
run:
Holiday
Party
XV
BUILD SUCCESSFUL (total time: 0 seconds)
```