**Report for Java Builder Pattern Project**

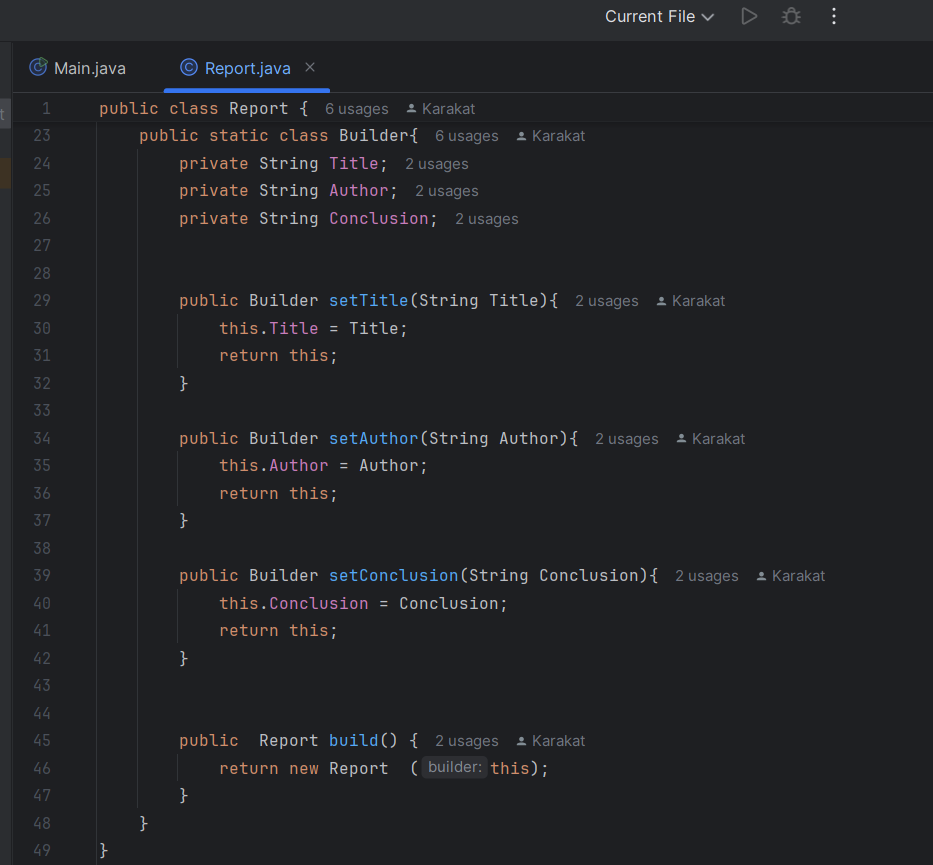
**Title: Report Builder Example**

**Author: Karakat Ibraim**

**Group: SE-2433**

**This Java project demonstrates the Builder design pattern using the Report class. The class has three private final fields: Title, Author, and Conclusion. To create Report objects in a flexible and readable way, a nested static Builder class is used.**

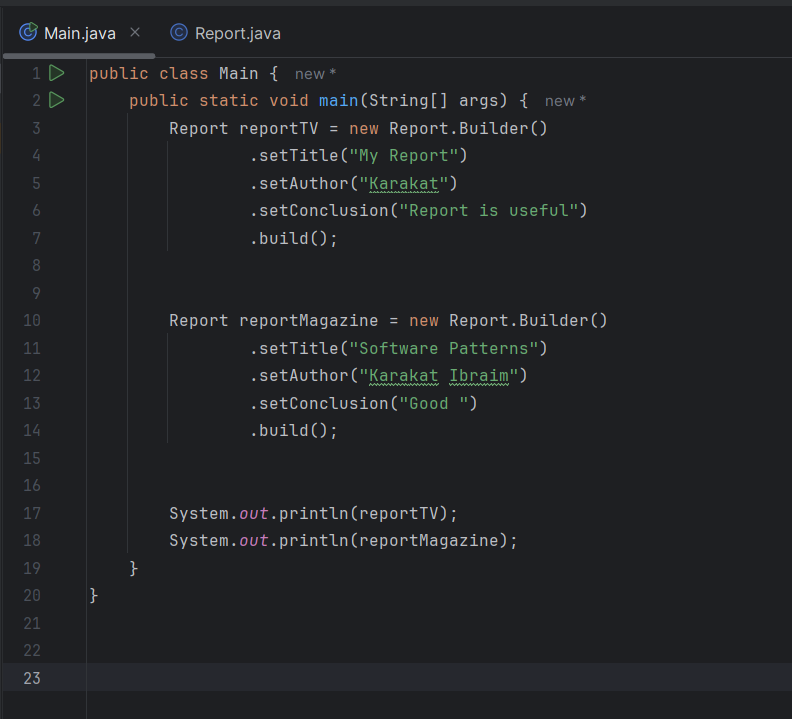
**The Builder class allows setting each field individually through methods (setTitle, setAuthor, setConclusion) and returns the Builder itself for method chaining. Once all desired fields are set, calling build() creates a new Report object with the specified values.**



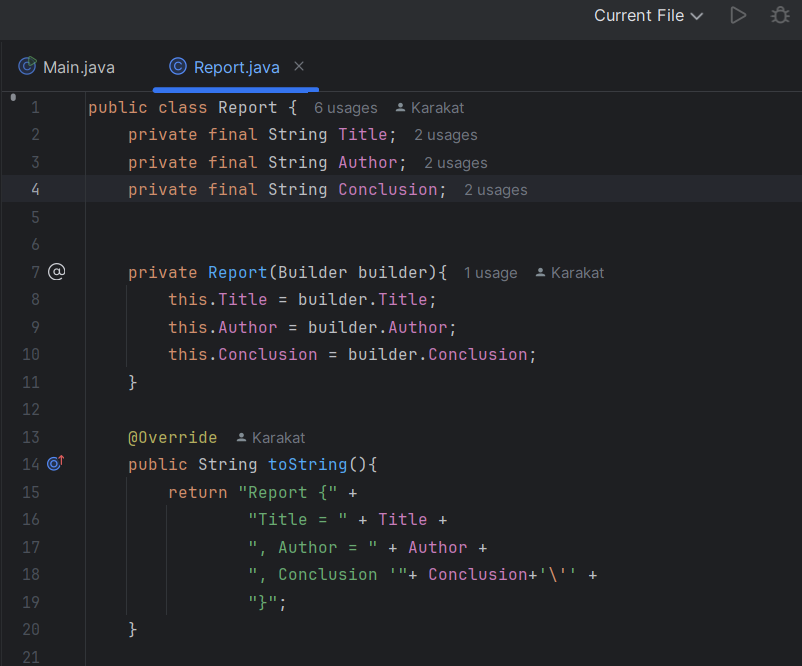
**In the Main class, two Report objects are created:**

**reportTV – Title: "My Report", Author: "Karakat", Conclusion: "Report is useful".**

**reportMagazine – Title: "Software Patterns", Author: "Karakat Ibraim", Conclusion: "Good".**



**The program prints both objects using the overridden toString() method, which formats the report fields clearly for easy reading.**



**Key Points:**

**The Builder pattern allows creating objects with multiple fields without writing multiple constructors.**

**Report objects are immutable, meaning their fields cannot be changed after creation.**

**Method chaining improves readability and makes code concise.**

**The toString() method provides a human-readable representation of the objects.**

**Conclusion:**

**This project successfully demonstrates how the Builder pattern can be used in Java to create complex objects in a clean, readable, and safe way.**