

Spring Professional Exam Tutorial v5.0

Question 03

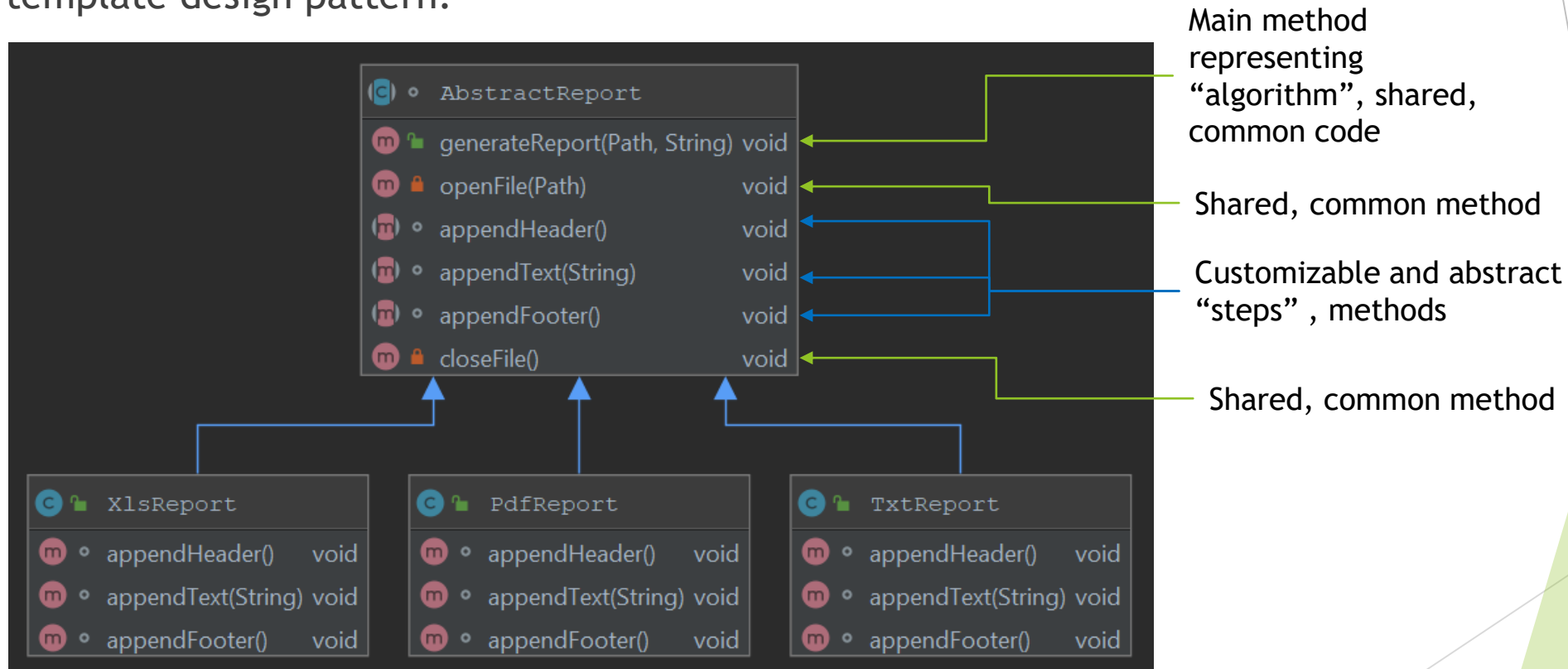
Question 03 - What is the Template design pattern and what is the JDBC template?

Template design pattern is a behavioral design pattern that can be used to encapsulate algorithm/main flow with its steps in a way to achieve steps customization and shared code reusability. It is achieved by creating abstract class that contains algorithm definition/main flow with shared code, and child classes extending abstract class which are customizing step or steps of the algorithm.

Template design pattern can be used to achieve greater code reusability, however since it is using inheritance, which is a very strong relationship between classes it can limit future flexibility of the system. You should use this pattern with caution and you should analyze if strategy design pattern will not give you similar results. Strategy uses composition instead of inheritance and in some cases instead of using template method, strategy can be used to achieve code reusability and also code flexibility.

Question 03 - What is the Template design pattern and what is the JDBC template?

Below you can find class diagram for report subsystem written with usage of template design pattern.



Question 03 - What is the Template design pattern and what is the JDBC template?

Jdbc Template is a class located in `org.springframework.jdbc.core` package. Goal of this class is to simplify use of JDBC by providing implementation of JDBC workflow, leaving application to provide SQL statements and results extractions.

Jdbc Template executes SQL queries or updates, initiates iteration over `ResultSet`, `ResultSet` mapping, also it catches exceptions and translates them into generic exceptions.

Code that interacts with Jdbc Template needs to provide implementation of callback interfaces which allows specific steps of JDBC workflow customization:

- ▶ `PreparedStatementCreator`
- ▶ `ResultSetExtractor`
- ▶ `PreparedStatementSetter`
- ▶ `RowMapper`

