

CHECKPOINT #3

CHECKPOINT ESTIMATED TIME: 4 hours

WHAT YOU SHOULD DO:

1. Read short checkpoint plot.
2. Read user stories.
3. Read technical details.
4. Clone initial repo
5. Create database and required tables - use structure from the erd diagram located in technical details section
6. Fill your existing db with sample data from .csv file. You can do it in any way you want.
7. Plan your workflow!
8. Start coding
9. Remember about git. :) There is always a chance that someone will steal you internet. If we can't get your code, we cannot check it

BEWARE:

If you decided to work within IntelliJ IDE, remember to add proper data to git ignore! :)

JERZY'S HUMBLE HOME LIBRARY

STORY

Jerzy is really into new technologies. He's amazed how JERZYBOT works – you did a great job with its implementation! Now, Jerzy has appetite for more – his home book collection is really big and still expanding, so he asked you to create simple book management system to maintain all the stuff laying on his dusted shelves.

USER STORY:

1. As Jerzy I would like to add new book to my book collection
2. As Jerzy I would like to edit given book's data.
3. As Jerzy I would like to delete book from collection
4. As Jerzy I would like to search for a books by one of theirs parameters (by ISBN number, Title, author, publication year, publisher's name)
5. As Jerzy I would like to see all books available in library sorted ascending by name of books
6. As Jerzy I would like to see all books written by given author

EXPECTED OUTPUT:

1. Program is simple console application
2. Program will print simple menu with all available options based on user story
3. Program is foolproof.
4. Program is sql-injection-proof

ADDITIONAL USER STORY:

Jerzy is eager to pay even more coolcoins for additional features! Before implementation make sure that all basic features works fine! (and all the sql queries are injection-proof!):

1. As Jerzy I would like to see how many books authors created.
2. As Jerzy I would like to see all books written in the last 10 years.
3. As Jerzy I would like to see which of the books is the most expensive one.
4. As Jerzy I would like to display full name of the author and his/her age

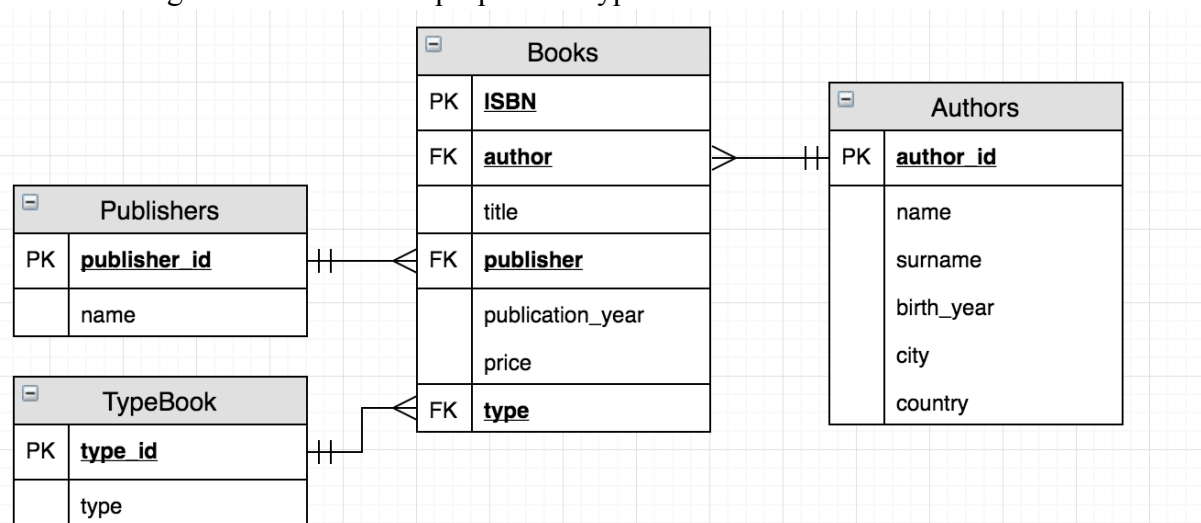
TECHNICAL DETAILS

CODE ARCHITECTURE

1. **You must implement** program within **MVC structure**.
2. **Use DAO** pattern for data access.
3. **Implement Iterator** pattern to customize iteration over books collection (Optional?)
4. **Divide code into proper packages**

DATABASE ARCHITECTURE

1. During tables creation use proper data type.



OOP

1. Use **inheritance** within your project structure.
2. Use **polimorphysm** for the sake of flexibility and data hiding
3. Use **encapsulation**.
4. Remember about **abstraction**!

CLEAN CODE

1. Do not forget about basic principles (DRY etc), follow java code guidelines

WHAT WILL BE EVALUATED?

1. Quality of code structure (Architectural patterns application)
2. Clean code
3. OOP principles usage
4. User story implementation