

Epilogue

In this book, you have learned most of the key foundational design patterns. I call them foundational because these are the classic design patterns that inspired all the other design patterns that you may encounter.

And there was a great value in learning these design patterns. Maybe some of them are almost obsolete. For example, real-life situations where using the classic version of the Singleton design pattern are extremely rare. But the principles behind each of these design patterns still live on.

You may no longer be able to find an appropriate scenario to use the Bridge design pattern in its classic form. But still, this pattern made other, more modern design patterns possible. For example, both Model-View-Controller and Model-View-ViewModel design patterns are based on Bridge. Likewise, Bridge has inspired the modern architecture of distributed applications, where the UI and the back-end API are represented by different applications.

And that's a good example of why I call these design patterns "foundational". Knowing them makes it easy to learn any other design patterns. And it makes it easy to understand many of the modern-day architectural best practices.

So, don't be put off by the fact that you haven't seen things like Model-View-Presenter or Repository in this book. There are many design patterns out there and it's just impossible to cover them all in a single book. But now, since you know the most fundamental ones of them all, you will have absolutely no problem learning any other design patterns that have been invented more recently.

There is only one classic design pattern that we haven't included in this book. And it's called Iterator. But there was a good reason

not to include it. First of all, in my entire career both writing commercial software and looking at open-source code, I haven't seen a single example where this design pattern was being used. Not one. Secondly, this design pattern is fairly complex and adds very little value compared to other patterns. Therefore the decision was made to leave it out.

A quick reminder that, if you want to get in touch with me regarding the content of the book, you can contact me via either Twitter or LinkedIn. Perhaps, there is additional content that you want to have included in the next edition of the book. Or maybe you found some errors in the current edition. Any feedback is welcome. And this is how you can get in touch:

X (Twitter): <https://twitter.com/FSazanavets>

LinkedIn: <https://www.linkedin.com/in/fiodar-sazanavets/>

Also, if you found this book useful, please write a review about it on the site you bought it from. This will help me as the author and will help the algorithms to show this book to more people who might need help with learning design patterns.

And, of course, if you liked this book, spread the word. Let your friends know about it. This way, you will help more people to become better software developers.

All the best and good luck!