

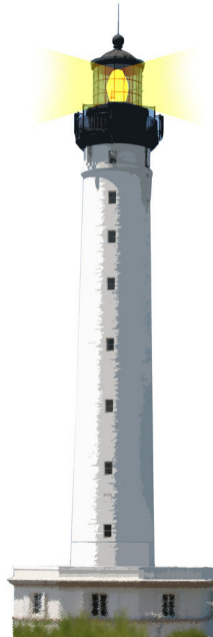
Advanced Object-Oriented Design

Advanced Object-Oriented Design

S. Ducasse and G. Polito



<http://www.pharo.org>



Lectures Web Site

- `https://github.com/UnivLille-Meta/Miage22`
- All slides are at
`https://rmod-files.lille.inria.fr/?dir=DesignCoffeeClub/Slides`



Testimonies

- "J'ai trouvé ça très intéressant, beaucoup plus que prévu ! je regrette de ne pas m'y être mis plus tôt. J'ai enfin l'impression de vraiment faire de la POO ! Ou à l'inverse je me rend que je n'en faisais pas vraiment... - Anonymous, 2019"
- "En fait on peut apprendre facilement plein de trucs car on peut voir tout le code." Abir 2022
- "I finally understood objects after five years of Java and working in a C++ company."
- "I have already learned so much! I have spent the last 20 years or so in software development and, following this Mooc, I realized I hadn't really grasped the essence of object-oriented design. - Anonymous"



Yes it can feel strange but

- First exercise: Watch Karate Kid I (the original movie)



<https://www.youtube.com/watch?v=SMCsX19SGgY>

- and do what you are told to do

Our main goal as OO experts

- Make sure that you will **never program the same** after this lecture!
- In **any** OOP languages!



Multiple Goals

- Advanced Object-Oriented Design
- Tests and TDD
- Reverse engineering
- Reflect on Learning
- Improve your communication skills
- Work on real unknown code



Advanced Object-Oriented Design

- **Revisit** and realign OOP concepts
- **Advanced** topics
 - double dispatch
 - design patterns
 - advanced patterns
- **Real world** case studies
 - Regis (a conference registration server)
 - Pillar (book compilation chain)
 - Iceberg (git tool)
 - Citezen (bibliographic library)
 - Bloc (graphical library)



Tests...

- You know tests are important
- Why don't you use them?
- Because you did not feel the **pain**
 - Is my change breaking something?
 - If I merge this change does it work?



Reverse engineering

There is 80% of chance that you will work on programs that you did not write

- How to understand fast the architecture?
- Navigate/browse large code bases
- Understand design
- Identify key points



Some 'fun' stories about real business

- Half of the software of a large Belgium major insurance company was written in a language that does not appear on google!!!
- They got 700,000 Euros fine. Only that :)
- A bank in Paris is running all its business on an orphan language mixing COBOL and working on an obsolete OS that is simulated.
- Others are a mixture between databases and Pascal where we had to wait for a recent version to be able to save the program in a file!
- Key energy-related business in France is still using Fortran 77 with **proprietary** extensions.



A pakistanese Javascript dev ...

- is paid 8 Euros per hour!
- Are you sure that you want to compete?
- Or do you want to sell your capacity to learn and focus on the added value?



About attitude

- You may complain like a lemming that "there is no documentation"...
- You may better
 - Learn how to learn fast
 - Learn how to jump over obstacles
 - Develop skills that are not mainstream



About learning

- How do you learn?
 - visual, oral?
 - repetition?
 - mental map?
 - iteration?
- What are your strategies to learn fast?
- What are your strategies to improve the quality of your learning?
- We will ask for a presentation on that!



The case of Pharo

- <http://mooc.pharo.org>
- Discord channel <https://discord.gg/QewZMZA>
- <http://books.pharo.org>
- Mailing-lists



Reporting

- Pull vs. Push feedback
- Good reporting lowers manager stress
- Weekly report will be part of the grade!

About soft skills

Soft skills (communication, autonomy, networking, teamwork)

- Soft skills are not well taught at university.
- Let us practice them.
 - How to help other people?
 - How to ask for help so that someone that you do not know and that does not earn anything will still help you?

Practice, learn, practice

- reflect on your process
- practice again



Grading

- Organize in groups of 3
- Register and create your own folder at <https://github.com/UnivLille-Meta/Miage22>
- You have the right to kick out someone not working from your group!
 - read resources about the Mollasson

Grading

- Two presentations (reverse engineering and design + actions)
 - Week 4 and 12
- Two exams: week 6 and week 9
- Work done as exemplified by Pull Requests
- Short weekly report showing
 - what has been done
 - what's next
 - actions to learn (discord chat, watching videos)
 - actions to help others
 - PR



Schedule

- 1 Week 8/9 Sep
- 2 Week 15/16 Sep
- 3 Week 22/23 Sep
- 4 Week 29/30 Sep - **Presentations**
- 5 Week 6/7 Oct
- 6 Week 13/14 Oct - **Examen**
- 7 Week 20/21 Oct
- 8 Week 27/28 Oct
- Ferie Week 10/11 Nov (Pay attention no lecture this week.)
- 9 Week 17/18 Nov - **Examen**
- 10 Week 24/25 Nov
- 11 Week 1/2 Dec
- 12 Week 8/9 Dec - **Presentations**



Admin

- `https://github.com/UnivLille-Meta/Miage22` pour le cours
- All slides are at
`https://rmod-files.lille.inria.fr/?dir=DesignCoffeeClub/Slides`



Resources

- Mooc <http://mooc.pharo.org>
- Discord <https://discord.gg/QewZMZA>
- Books: <http://books.pharo.org>
- Git:
 - <https://rmod-files.lille.inria.fr/DesignCoffeeClub/Slides/Support-1-BasicGit.pdf>
 - <https://rmod-files.lille.inria.fr/DesignCoffeeClub/Slides/Support-2-AdvancedGit.pdf>
 - <https://rmod-files.lille.inria.fr/DesignCoffeeClub/Slides/Support-2-PracticalGit.pdf>
- Agile programming Explained
- Design patterns
 - Smalltalk design pattern companion
 - Gang-of-four book (Gamma et al)
- Pharo by Example <http://books.pharo.org>
- Pharo with Style <http://books.pharo.org>



Conclusion

- A lecture with multiple objectives
- Remember you can learn a lot!



A course by

S. Ducasse, G. Polito, and Pablo Tesone



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>