April 5, 2024

DARTMOUTH LIBRARIES

Classy Code: Object-Oriented Programming for Fun and Profit

RAUX Training

Simon Stone
Research Data Services
Dartmouth College Libraries

Intro
What is Object-Oriented Programming (OOP)?

- Programming paradigm (cf. “procedural” or “functional”)
- Based on the concept of “objects”
- Objects contain data and procedures to do things with that data
- Structure of a program is expressed in the interactions between the objects:
  - “The ‘DataPreprocessor’ uses the ‘FileReader’ to import the measurements”

Intro
Why bother?

OOP can help to increase:

- Modularity of code for easier testing and troubleshooting
- Reuse of code within and across projects
- Expressiveness of the code by using real-world analogies
- The portability of code between platforms or languages

Intro
Where can I do OOP?

Most modern programming languages offer some form of support for OOP in addition to other paradigms (*multi-paradigm*):

- C++, C#, Java, JavaScript, MATLAB, PHP, Python, R, Swift, …

Even if you are not going to use OOP yourself, the frameworks you use probably do!

Intro
Agenda

- Objects and classes
- Four key concepts:
  - Encapsulation
  - Inheritance
  - Composition
  - Polymorphism
- Design patterns
- Summary and next steps

Let’s get classy…

Hands-on

Outro
Key take-aways

- Think about your problem in terms of **objects** and their **interactions**
- Encapsulation: Keep data and methods on that data together in a class
- Inheritance: Model an ***is a*** relationship between classes
- Composition: Model a ***has a*** relationship between classes
- Polymorphism: Use the same interface or symbol with different objects
- Design patterns: If classes are atoms, design patterns are molecules

Outro
Next steps

- Rewrite some of your existing code using OOP!
- Study SOLID design principles
- Look at more design patterns
- Write, write, write!
- Read, read, read!
- Checkout further tutorials, e.g., https://refactoring.guru/design-patterns