DAY 2

OO Design - classes and relationships - slide 16

Review homework – what classes, attributes and methods do we need for the library system?

Associations - and how many objects are on each side
Aggregations
Compositions
Classes
Heirarchies

CRC exercise - slide 19

Using the library lending system, assume a role and execute a scenario.
Possible classes are:

- application
- library
- material
- books
- DVDs
- journal
- company
- system
- ID numbers
- item
- borrower
- date
- organization
- employee
- article
- fine

but there may be duplicates or some outside the scope of the system. Which are actors?

Your scenarios are:

- What happens when Johnny Codewarrior, who has no accrued fines and one outstanding book, not overdue, checks out a book entitled "*Document, Your job depends on it*"?
- Judy Hacker, has fines of £2 outstanding, would like a DVD "*Debugging to music"*, does have a book out (not overdue) and is bringing back an overdue journal.
- Miss Marple wants to borrow a journal "*Sleuthing in C#"* but can't find it… is it already out?
- Eric Halfbee comes in with a pile of overdue items, but doesn't know if he has enough money to pay off his debts. If he has, he'd like a borrow a DVD.

Drawing structure and behaviour – slide 21

Sketch out the structure of the classes in the library system and their relationships. Hint: some may be built-in classes.

Pick one of the scenarios above and draw a sequence diagram to represent the behaviour of all the objects involved.

Implementation - slide 24

Create an empty PyCharm project from the cloned GitHub repo. ready to code up the Library System – good practice would suggest a folder (or Python Package as appropriate) for each of source code, test code and documentation.

Each person should take responsibility for one class and sketch it out in a skeleton code Python file. The approach is to define your class and its methods and attributes using docstrings and comments and NO ACTUAL CODE to start with.

After discussion with the team to ensure we are following the design, each person should implement their class methods: we should concentrate on one scenario / sequence diagram.

Don't forget to use GitHub so that we work as a team ☺

Naming - slide 26

Carry out a code review in pairs and look for good coding practice generally and especially the OO naming conventions used. Do they adhere to standards? Is the code readable? Are there docstrings with the ethod arguments explained? Are there comments?

HOMEWORK

Ensure you can share code with a GitHub account and revise its use if necessary. Hook up your PyCharm project into your git repository cloned from the shared one (if not already done); add, commit and push your skeleton code files, and pull the ones made by your colleagues.

Carry on implementing the methods!