Méthodes de Test
TP 3 — Mock Objects

Stefano Zacchiroli
zack@pps.univ-paris-diderot.fr

Laboratoire PPS, Université Paris Diderot

2013–2014

Challenge #1 — design pattern testing

1. pick 2 or more design patterns from the "Gang of Four" book, preferring patterns that implement reasonably complex communication protocols between objects
   - most behavioral patterns are good choices

Gamma, Helm, Johnson, Vlissides
Design Patterns: Elements of Reusable Object-Oriented Software.
Addison-Wesley, 1995.

2. ask me to validate your choices of design pattern
3. implement the patterns
   - using TDD or not, that's up to you
4. provide a convincing test suite that uses object mocking to show that your pattern implementations adheres to the expected communication protocols for the chosen patterns

Challenge #2a — network service testing

Either:

1. pick an Internet application layer protocol, see for example the list at `http://en.wikipedia.org/wiki/Application_layer`
2. ask me to validate your choice of protocol
3. [if you don't know about them] learn the basics about it and its messages
4. model, in an object-oriented language, clients and servers for the chosen service
5. provide a convincing test suite that uses object mocking to show the expected behavior of client/server implementations
6. implement some client/server functionalities, just enough to make some tests pass

Challenge #2b — network service testing (cont.)

Or:

1. choose a network service used for communication, e.g.:
   - Twitter / status.net / pump.io
   - Jabber / XMPP
   - Gtalk / Facebook chats / IRC
   - Google Hangout / Skype / SIP
   - Titanpad / Etherpad / Google Drive
   - ...
2. model, in an object-oriented language, clients and servers for the chosen service
3. provide a convincing test suite that uses object mocking to show the expected behavior of client/server implementations
4. implement some client/server functionalities, just enough to make some tests pass

Homework 2 — description and submission

Description

- work in pairs
- win against challenges $(\#1 \wedge (\#2a \vee \#2b))$

Submission

- submit a `tar.gz` archive containing
  - a README documenting
    - the chosen exercises
    - the members of the pair
    - instructions to: compile and run the test suite
  - the source code of your work
- submissions must be made via DidEL at:

  `http://didel.script.univ-paris-diderot.fr/claroline/course/index.php?cid=METHTEST`

  Section: *Travaux* > *TP3 — mock objects*
- deadline: 18 November 2013 (2 weeks from now)