|  |  |
| --- | --- |
| D:\PAL\Logo_4_21_15\Logo_4_21_15\Primary Logo\png_files\EPAM_LOGO_Full_Color_RGB.png | CDP Automated Testing Mentoring Program  2017-Q2 |

Module 5.2: Code Design Patterns in Test Automation (Java)

## home task

1. Implement the following design patterns in your solution from previous module\*:
   1. Singleton
   2. Factory Method
   3. Decorator

You can use any area of you code to apply the pattern (any code layer – test, service or page object and their combinations).

You may consult with your mentor about the specific area of applying the pattern as well, but try to make a self-dependent final decision.

1. Revise your code to match S.O.L.I.D. principles. Fix Provide list of corrections to your mentor, e.g.:

|  |  |  |
| --- | --- | --- |
| Class | Problem | Solution |
| FactoryDriver | Open-closed principle is broken.  Method getCurrentDriver has public access,which can renew driver instance without passing the assertion | Accessor of method getCurrentDriver is set private. |
| GmailPage | Open-closed principle is broken.  Method buildSubjectString is public, although it is used inside the class. | Accessor of method buildSubjectString is set private. |
| GmailPage | Class single responsibility is broken.  Class does Actions | All usage of actions migrated to standalone class GActions |
| BaseTest, GmailTest | Class single responsibility is broken.  These classes interact with PO layer, but suppose with PS layer. In case developer amend PO layer functionality,that affects both – Test layer and PS layer | Encapsulate all PO interactions in PS layer, thus Test layer interacts with PS layer only. |

## Bonus TAsk

Implement any extra pattern mentioned during the training session (for extra mark) or any other pattern instead of the list above. Exact pattern list should be discussed with your mentor.

## acceptance criteria

1. **PATTERNS**: All patterns from mandatory part (Singleton, Factory Method, Decorator) should be implemented.
2. **PATTERNS**: Classes which were modified/created during pattern implementation should be invoked during the test run. Just storing them in project packages is NOT enough.
3. **S.O.L.I.D.:** Code is revised to match S.O.L.I.D. principles.
4. **S.O.L.I.D.: at least** 3 fixes are described within the table per example above and delivered to a mentor.
5. **S.O.L.I.D.:** fixes are implemented in code structure.
6. **BONUS TASK:** extra patterns mentioned during the training session are implemented considering criteria from point #2 above.

***[\*] discuss assignment specifics with your mentor***