**Báo cáo Homework 7**

**Name : Brak Lihou – 20200836**

**Module được chọn:**

* Phương thức getWorkDetails(LocalDate date) của lớp WorkDetail cho Worker
* Phương thức getWorkDetails(LocalDate date) của lớp WorkDetail cho Officer

**Tài liệu mô tả phương thức/lớp**

Phương thức getWorkDetails(LocalDate date) trả về một đối tượng WorkDetail cho một ngày làm việc cụ thể. Đối tượng WorkDetail chứa các thông tin về ngày làm việc, ca làm việc, số giờ đi làm muộn, số giờ nghỉ sớm.

**Phân tích kỹ thuật kiểm thử**

* **Kỹ thuật kiểm thử hộp đen**

Dựa trên yêu cầu chức năng của phương thức getWorkDetails(LocalDate date), chúng ta có thể thiết kế các test case sau:

|  |  |  |
| --- | --- | --- |
| Tên test case | Đầu vào | Đầu ra mong đợi |
| TC01 | Ngày làm việc hợp lệ | Đối tượng WorkDetail chứa các thông tin hợp lệ |
| TC02 | Ngày làm việc không hợp lệ | null |
| TC03 | Ca làm việc không hợp lệ | null |
| TC04 | Số giờ đi làm muộn không hợp lệ | null |
| TC05 | Số giờ nghỉ sớm không hợp lệ | null |

* **Kỹ thuật kiểm thử hộp trắng**

Dựa trên độ đo c1, chúng ta có thể thiết kế các test case sau:

|  |  |  |
| --- | --- | --- |
| Tên test case | Đầu vào | Đầu ra mong đợi |
| TC06 | Ngày làm việc là ngày thường | Ca làm việc là ca sáng hoặc ca chiều |
| TC07 | Số giờ đi làm muộn là số nguyên dương | Số giờ đi làm muộn không vượt quá giới hạn cho phép |
| TC08 | Số giờ nghỉ sớm là số nguyên dương | Số giờ nghỉ sớm không vượt quá giới hạn cho phép |

**Chương trình kiểm thử tự động**

Chương trình kiểm thử tự động được viết bằng JUnit framework. Các test case được viết trong các class con kế thừa từ class TestCase.

**Các Class Kiểm thử như sau :**

* Employee.java
  + - các hàm Testcase trong lớp nay là testEmployeeConstructor(), testGetterSetter()
* Officer.java
  + - các hàm Testcase trong lớp nay là testConstrutor\_ValidArguments()
* Worker.java
  + - các hàm Testcase trong lớp nay là testWorkerInheritsEmployee(), testWorkerConstructor() ,
* WorkDetailTest.java
  + - các hàm Testcase trong lớp nay là testConstructor\_WeekdayWithValidWorkShifts(),testConstructor\_InvalidWorkDate(),testConstructor\_InvalidWorkShifts(),testConstructor\_InvalidLateHours(),testConstructor\_InvalidEarlyLeaveHours(),\testConstructor\_WeekendWithRestShift(),testConstructor\_ValidPositiveLateHours(),testConstructor\_InvalidLateHoursExceedingLimit()