Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC3745 — Testing
2023 - 1

Tarea 4

Fecha de entrega: Martes 25 de Abril del 2023 a las 23:59

Información General

La siguiente tarea contempla como objetivo principal aplicar en un ejemplo sencillo las técnicas y herramientas asociadas a las pruebas de unidad: statement coverage, mutation testing, y test smells.

Objetivos

- Aplicar los conceptos de coverage para crear una batería de pruebas que cubra todos los statements del código bajo prueba.
- Aplicar los conceptos de mutation testing para analizar la calidad de los assertions (oracles) de nuestra bateria de pruebas.
- Utilizar un generador de pruebas de unidad automático y entender las limitaciones del estado del arte.

Tarea

Descargue el **proyecto clock display**(código base) disponible en Canvas y realice las siguientes actividades descritas a continuación:

- **2 pts. Actividad 1:** Crea un conjunto de casos de prueba de tal forma que se obtenga 100% statement de coverage. Para ello, genere el coverage report en html de su código. En caso de no lograr el coverage pedido se asignara un puntaje parcial según el porcentaje logrado.
- **2 pts. Actividad 2:** Su conjunto de casos de prueba debe ser capaz de matar a todos los mutantes que genere la librería mutatest. Podrían existir mutantes que no tenga sentido matar en ese caso deberán argumentar brevemente en el pdf a entregar porque no tiene sentido considerar ese mutante, si la justificación tiene sentido no se descontara puntaje por ese caso. Incluya en el pdf un screenshot como evidencia de la cantidad de los mutantes que mueren. Se asignará el puntaje según la cantidad de mutantes que logren matar y en caso de no sea capaz de matar a todos, se considerara la claridad de los motivos descritos.
- **2 pts. Actividad 3:** Genere tests unitarios utilizando la librería pynguin para el módulo `display_number` y `clock_display`. Escriba un párrafo sobre los principales test smells que encontró en el código generado, señale tambien algunos ejemplos en donde es posible observar los tests smells en el código. Ejemplo del comando en terminal para el modulo `display_number`:

```
export PYNGUIN_DANGER_AWARE=TRUE
pynguin --project-path ./ --output-path ./result --module-name src.display_number -v
```

Entregables

Se debe entregar un .zip con los siguientes archivos:

- El proyecto de canvas junto con los test creados para lograr el 100% de statement coverage.
- El reporte html generado por el modulo coverage de python.
- Un pdf que contenga el screenshot con el resultado de mutatest y el párrafo pedido sobre los test smells. La extensión **máxima es de una plana, tamaño de letra 12**. Se excluye del limite de extensión si incluyen alguna justificación con respecto a la actividad 2.
- Un archivo README que contiene los nombres de quienes realizaron la tarea y su aporte.

Reportar problemas en el equipo

En el caso de que algún integrante no aportara como fue esperado en la tarea, podrán reportarlo enviando un correo con asunto **Problema Equipo {NumeroGrupo} Testing** a *juanandresarriagada@uc.cl* con copia a *afernandb@uc.cl* explicando en detalle lo ocurrido. Posterior a eso se revisara el caso en detalle con los involucrados y se analizara si corresponde aplicar algún descuento. Instamos a todas las parejas que mantengan una buena comunicación y sean responsables con el resto de su equipo para evitar problemas de este estilo.
Advertencia: Si algun integrante del grupo no aporta en las tareas puede implicar recibir nota mínima en esa entrega.

Restricciones y alcances

- Su programa debe ser desarrollado en **Python 3.9 o superior**.
- Los archivos de código entregados deben terminar con la extensión **.py**.
- En caso de dudas con respecto al enunciado deben realizarlas en un foro relacionado a la tarea que se encontrara disponible en canvas.
- Si no se encuentra especificado en el enunciado, supón que el uso de cualquier librería de Python adicional a las utilizadas en el código base se encuentran prohibidas. En caso de que estimes necesario podrás preguntar en el foro de la tarea por el uso de alguna librería adicional.

Politica de atraso

Se efectuara un descuento por entregar tareas atrasadas. Se descontara 0.5 si la tarea se entrega con menos de una hora de retraso. El puntaje final en caso de atraso seria calculado mediante la siguiente formula:

$$PuntajeFinal = PuntajeObtenido - (0,5 + 0,05 \cdot k)$$

, donde k es el numero de horas de retraso menos uno.

Integridad académica

Este curso se adscribe al Código de Honor establecido por la Escuela de Ingeniería. Todo trabajo evaluado en este curso debe ser hecho **individualmente** o en **los grupos asignados** según sea definido en la evaluación y **sin apoyo de terceros**. Se espera que los alumnos mantengan altos estándares de honestidad académica, acorde al Código de Honor de la Universidad. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada alumno conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería.

¡Éxito con la Tarea! :D