PROGRAMACIÓN ORIENTADA A OBJETOS

## Excepciones

## Laboratorio 4/6

## Andrés Camilo Oñate Quimbayo, Nicolas Ariza Barbosa

## Activity

## EN BLUEJ

## PRACTICANDO MDD y BDD con EXCEPCIONES

## 1. Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.

## El proyecto no compila porque en métodos de las tres clases disponibles y la prueba unitaria

## 

## Existen métodos que nos arrojan excepciones sobre posibles fallos

## 

## 

## 

## 

## Y la razón por la que no compilan es porque no hemos definido aún la clase de excepciones para este proyecto y al no encontrarla no puede asegurar las excepciones que buscamos prever.

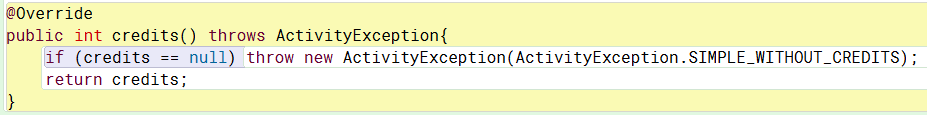
**Realicen las adiciones necesarias para lograrlo.**

Definir una clase propia para las excepciones ActivityException, encargada de manejar las excepciones con métodos heredados de la clase Excepción, de Java.

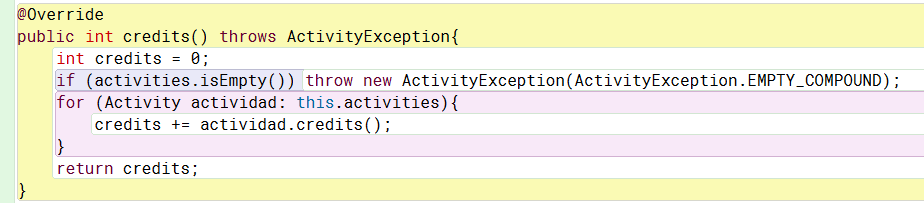
## 

2. Dado el diseño y las pruebas documenten y codifiquen el método credits().

Dado que la clase activity es abstracta, se debe codificar en cada una de las subclases el método credits(). En SimpleActivity se retorna el número de créditos que tiene por atributo, de acuerdo con el diseño se lanza una excepción si no hay créditos.

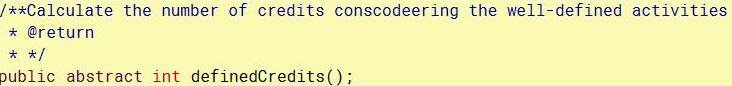


Para CompoundActivity, como cada una de las clases implementa este método, es suficiente con recorrer las actividades asociadas y hacer el llamado al método credits(), de acuerdo con el diseño se lanza una excepción si no hay actividades asociadas.



3. Dada la documentación, diseñen, codifiquen y prueben el método definedCredits().

Primero se menciona el origen de este método, el cual reside dentro de la clase abstracta *Activity* (En este lugar la acción referida no posee un cuerpo ya que se trata de un método abstracto)



Luego se sobrescribió dicho método en las clases que heredan el comportamiento, es decir, *SimpleActivity*

Texto

Descripción generada automáticamente con confianza media

Y *CompoundActivity*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ahora se generan las pruebas de unidad

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

A la hora de probarlas los resultados son correctos

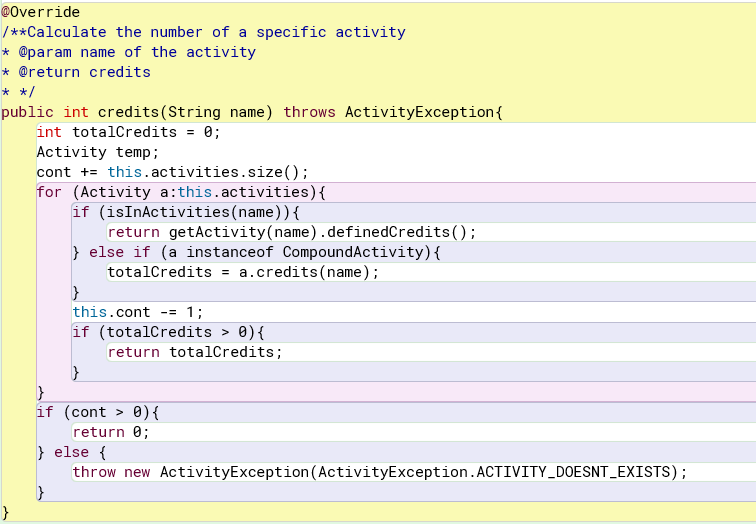
4. Diseñen, implementen y prueben el método credit(String).

En este método para calcular los créditos de una de las actividades vamos a incluir dos casos especiales: 1) no existe una actividad con esa descripción y 2) existen dos actividades con la misma descripción

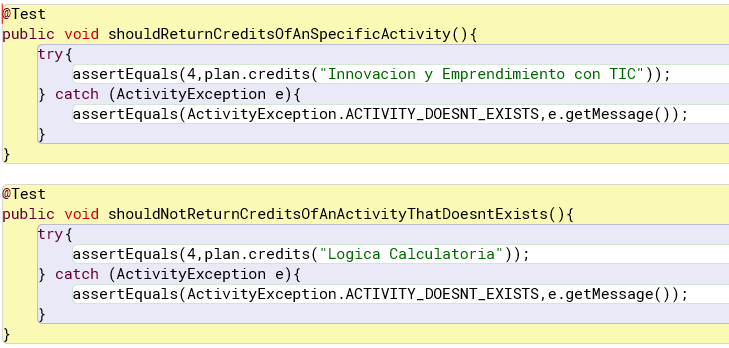
Implementando dos métodos nuevos



Se recorre recursivamente todas las listas de la actividad compuesta principal y a su vez se llama a las subactividades compuestas hasta encontrar la actividad buscada. En caso de no encontrarlo se generará la excepción correspondiente.



Se generaron dos pruebas de unidad



Y ambas funcionan correctamente

## CURRICULUM

## EN CONSOLA

1. En su directorio descarguen los archivos contenidos en curriculum.zip, revisen el contenido. ¿Cuántos archivos se tienen? ¿Cómo están empaquetados? ¿Cómo deberían estar organizados?

El archivo tiene 3 archivos con extensión .java, no están empaquetados y deberían estar en una carpeta con nombre src.

1. Revisen el diseño del programa: diagramas de paquetes y de clases. ¿cuántos paquetes tenemos? ¿cuántas clases tiene el sistema? ¿cuál es la clase ejecutiva?

Tenemos dos paquetes:

El sistema tiene un total de 6 clases

La clase ejecutiva es

1. Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener? ¿qué clases deben tener?

Se deben crear los tres directorios básicos:

Texto

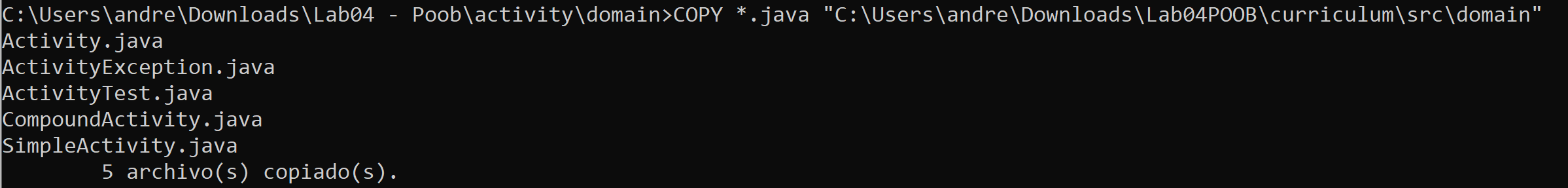
Descripción generada automáticamente

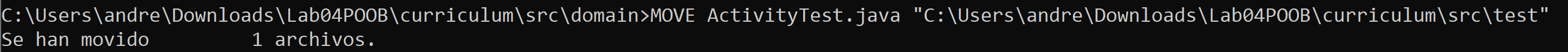
El directorio contiene subdirectorios que contiene el código fuente \*.java de cada uno de los paquetes.

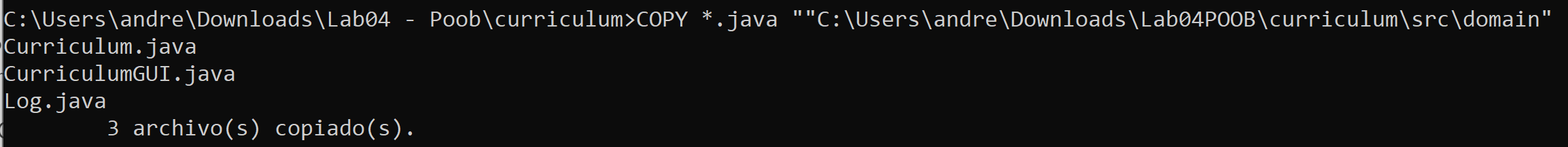
Texto

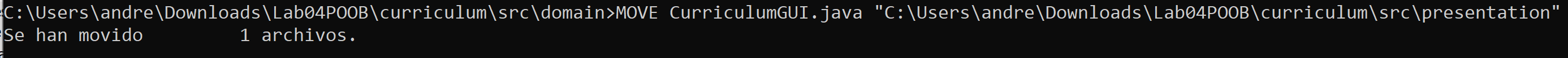
Descripción generada automáticamente

Se copian y mueven los archivos correspondientes:









Patrón de fondo

Descripción generada automáticamente con confianza baja

En src: En cada paquete se pone el codigo correspondiente.

Texto

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamenteImagen que contiene Texto

Descripción generada automáticamente

Generando documentación:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Ejecute el proyecto, ¿qué funcionalidades ofrece? ¿cuáles funcionan?

Se compila el proyecto:

Se ejecuta el programa:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* Permite Listar, adicionar o buscar un curso.
* Tiene dos botones: Listar, Limpiar.

Al presionar en Listar:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Botón Limpiar:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* Adicionar

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Buscar

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

1. ¿De dónde salen los cursos iniciales? Revisen el código y la documentación del proyecto. ¿Qué clase pide que se adicionen? ¿Qué clase los adiciona?

* Los cursos iniciales salen de la clase , se agregan con el método

.

* pide que se adicionen.
* los adiciona.

## Adicionar y listar. Todo OK.

El objetivo es realizar ingeniería reversa a las funciones de adicionar y listar.

1. Adicionen un nuevo curso

¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El curso aparece en la pestaña listar del programa:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Al presionar en adicionar no se da una respuesta grafica de permita verificar que se agregó correctamente el curso. El comportamiento no es adecuado, se debería notificar al usuario sobre el resultado de la operación.

1. Revisen el código asociado a la funcionalidad de adicionar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

* Capa de presentación: ,
* Capa de dominio:

1. Realicen ingeniería reversa para la capa de dominio para adicionar. Capturen los resultados de las pruebas de unidad.

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Revisen el código asociado a la funcionalidad de listar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de dominio?

Capa de presentación: ,

Capa de dominio:

1. Realicen ingeniería reversa para la capa de dominio para listar. Capturen los resultados de las pruebas de unidad.
2. Propongan y ejecuten una prueba de aceptación.

**Objetivo:** Validar las funcionalidades listar y adicionar de la aplicación.

**Características:**  El usuario puede adicionar y listar nuevos cursos desde las ventanas correspondientes de la aplicación.

**Acciones:**

* **Funcionalidad de adicionar:** Bajo las condiciones ideales el usuario ingresara los datos básicos de un curso: Sigla, Nombre, Créditos, Resultados de Aprendizaje en los espacios correspondientes, después debe dar un clic sobre el botón “Adicionar”. Es prudente que después de adicionar limpie el espacio con el botón “limpiar”.
* **Funcionalidad de listar:** Cuando el usuario quiera ver los cursos agregados por el momento (incluyendo los que están por defecto) presionara el botón “listar”.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Adicionar un curso. Funcionalidad robusta

1. El objetivo es perfeccionar la funcionalidad de adicionar un curso para hacerla más robusta.

**Para cada uno de los siguientes casos realice los pasos del 1 al 4.**

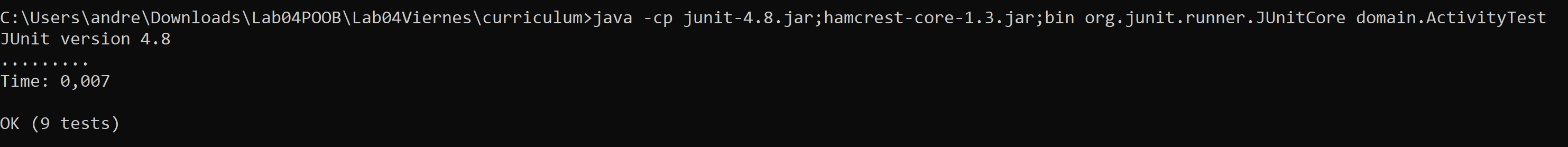
1. **¿Y si no da la sigla?**
2. Propongan una prueba de aceptación que genere el fallo.

Se adiciona el curso Teoría de la programación, como las siglas de los cursos son importantes para otros sistemas de la universidad se debe presentar al usuario un error notificando la necesidad de este.

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente

* Lanzar: El método addSimple de la clase
* Propagar: El método addSimple de la clase
* Atrapar: actionAdd de curriculum GUI

1. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.



1. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

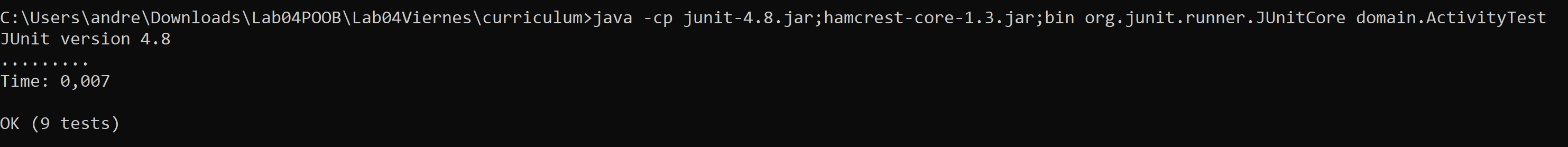
1. ¿Y si da una sigla de un curso que ya existe?
2. Propongan una prueba de aceptación que genere el fallo.

Se adiciona el curso TEORIA DE LA PROGRAMACIÓN con la misma sigla de un curso ya registrado, POOB, el sistema debe notificar al usuario que ya existe un curso con esa sigla.

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente

* Lanzar: El método addSimple de la clase
* Propagar: El método addSimple de la clase
* Atrapar: actionAdd de curriculum GUI

1. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.



1. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

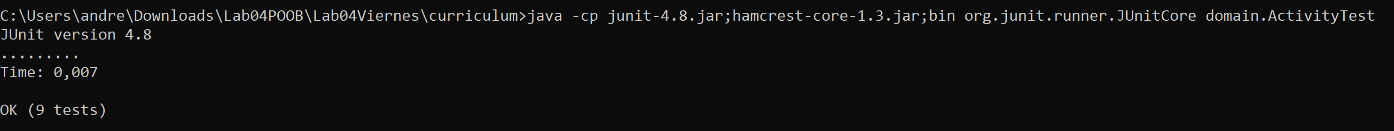
1. ¿Y si en créditos no da un número? ¿o no da un número entre [1..10]?
2. Propongan una prueba de aceptación que genere el fallo.

Se adiciona el curso el curso Teoría de la programación con sigla TPRO, y 3 créditos. En primera instancia se debe notificar no se permite ingresar el curso si no registra los creditos, al igual se valida su rango ingresando que tiene 11 creditos.

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente

* Lanzar: El método addSimple de la clase
* Propagar: El método addSimple de la clase
* Atrapar: actionAdd de curriculum GUI

1. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.



1. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

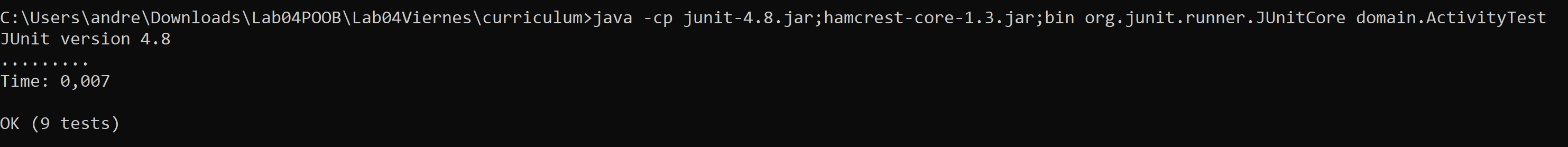
1. ¿Y si no indica el nombre de curso?
2. Propongan una prueba de aceptación que genere el fallo.

Se adiciona el curso Teoría de la programación con sigla TPRO, y 3 créditos. El sistema va a señalar que no se ingreso el nombre del curso.

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente

* Lanzar: El método addSimple de la clase
* Propagar: El método addSimple de la clase
* Atrapar: actionAdd de curriculum GUI

1. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.



1. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

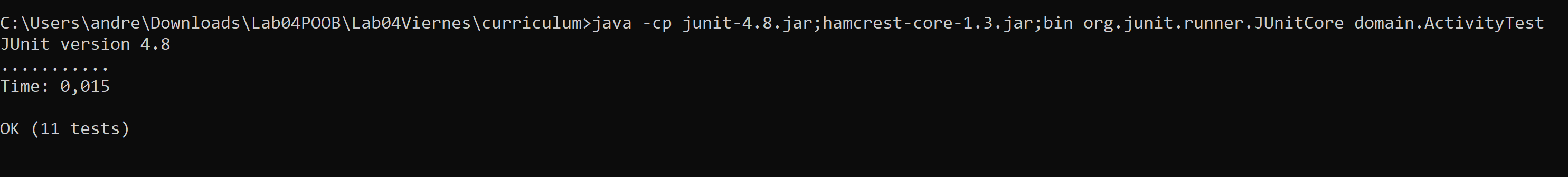
1. La descripción de los resultados de aprendizaje debe tener al menos 10 caracteres.

Se adiciona el curso Teoría de la programación con sigla TPRO, y 3 créditos. El sistema va a señalar que no se ingresó una descripción de los resultados de aprendizaje lo suficientemente larga.

1. Analicen el diseño realizado. Para hacer el software robusto: ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente

* Lanzar: El método addSimple de la clase
* Propagar: El método addSimple de la clase
* Atrapar: actionAdd de curriculum GUI

1. Construya la solución propuesta. Capture los resultados de las pruebas de unidad.



1. Ejecuten nuevamente la aplicación con el caso de aceptación propuesto en 1. ¿Qué sucede ahora? Capture la pantalla.

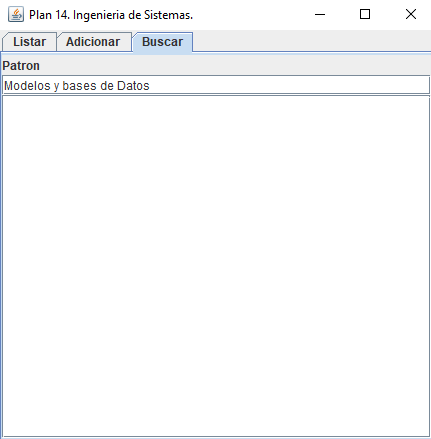
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Consultando por patrones.

## ¡No funciona y queda sin funcionar!

1. Consulten un curso especial que inicie con M. ¿Qué sucede? ¿Qué creen que pasó? Capturen el resultado. ¿Quién debe conocer y quien NO debe conocer esta información?



* ¿Qué sucede?

Es lanzado un NullPointerException.

* ¿Qué creen que pasó?

En algún momento de la ejecución se trató de llamar a algún método con una instancia de NULL.

* ¿Quién debe conocer y quien NO debe conocer esta información?

Esta información debe ser conocida por las clases implicadas, en este caso Curriculum y CurriculumGUI, y no debe ser conocida por otras clases como Activicty.

1. Exploren el método record de la clase Log ¿Qué servicio presta?

Trata registrar las excepciones generadas durante la ejecución para que el desarrollador sea capaz de mirarlas más adelante.

Texto

Descripción generada automáticamente

En caso de error, retornara una excepción y el recorrido que este hizo a través de todos los llamados.

Texto

Descripción generada automáticamente

1. Analicen el punto adecuado para que EN ESTE CASO se presente un mensaje especial de alerta al usuario, se guarde la información del error en el registro y continúe la ejecución. Expliquen y construyan la solución.

En el método se declarará un bloque try-catch, cada vez que ocurra una excepción de cualquier tipo, se mostrará al usuario un mensaje y el programa continuará con la ejecución, además de guardar el log un registro.

Texto

Descripción generada automáticamente

1. Ejecuten nuevamente la aplicación con el caso propuesto en 1. ¿Qué mensaje salió en pantalla? ¿La aplicación termina? ¿Qué información tiene el archivo de errores?

Salió en la pantalla un mensaje de error: "Ha ocurrido un error en la ejecución “

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

La aplicación no termina

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El archivo de errores tiene la información sobre la excepción, en este caso NullPointerException, además tiene en le encabezado información como la fecha y el paquete.

Texto, Carta

Descripción generada automáticamente

1. ¿Es adecuado que la aplicación continúe su ejecución después de sufrir un incidente como este? ¿de qué dependería continuar o parar?

La aplicación no debería detenerse sin que el usuario se lo diga, sin embargo, es adecuado que la aplicación le muestre un mensaje al usuario si debe detenerse por un error irrecuperable.

1. Modifiquen la aplicación para garantizar que SIEMPRE que haya un error se maneje de forma adecuada. ¿Cuál fue la solución implementada?

En el bloque try-catch del método , se agregará un nuevo caso el cual específicamente buscará la excepción n. Bajo cualquier otra excepción la aplicación terminará.

Texto

Descripción generada automáticamente

## Consultando por patrones.

## ¡Ahora si funciona!

1. Revisen el código asociado a la funcionalidad de buscar en la capa de presentación y la capa de dominio. ¿Qué método es responsable en la capa de presentación? ¿Qué método es responsable en la capa de dominio?

* Capa de presentación:
* Capa de dominio:

1. Realicen ingeniería reversa para la capa de dominio para buscar. Capturen los resultados de las pruebas. Deben fallar.

La prueba fallo:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

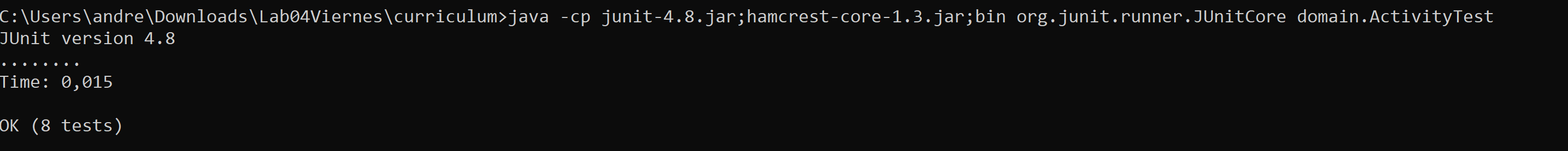
1. ¿Cuál es el error? Soluciónenlo. Capturen los resultados de las pruebas.

El error está en el método de clase , el ArrayList “resultados” no está bien instanciado.

Texto

Descripción generada automáticamente

Resultados de las pruebas:



1. Ejecuten la aplicación nuevamente con el caso propuesto. ¿Qué tenemos en pantalla? ¿Qué información tiene el archivo de errores?

Al probar con el caso propuesto “M” obtenemos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

El archivo de errores no se modifica en este caso, ya que deja de ocurrir la excepción.

1. Refactorice la funcionalidad para que sea más amable con el usuario. ¿Cuál es la propuesta? ¿Cómo la implementa?

Se implementa un método en la clase que devuelva el formato esperado, dado que en el método de se llama a un método toString de la clase Activity el cual no estaba implementado, lo que se mostraba al usuario era una referencia del objecto.

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

## Retrospectiva

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? 12 (Horas/Hombre)
2. ¿Cuál es el estado actual del laboratorio? ¿Por qué?

El laboratorio esta completo.

1. Considerando las prácticas XP del laboratorio. ¿cuál fue la más útil? ¿por qué?

Coding standards ha sido la practica XP más útil ya que esta nos invita mantener una escritura de código consistente y fácil de leer entre los dos a la hora de realizar cualquier tipo de avance. En caso de detectarse alguna falencia o mejora que se pueda realizar, la finalidad principal del código no se verá afectada por alguna clase de mal entendido.

1. ¿Cuál consideran fue el mayor logro? ¿Por qué?

Pasar de BlueJ a consola, realizar los cambios al código, ejecutar las pruebas y el programa, fue un cambio que no tenía contemplado.

Implementar el método *credits* el cual era un método de búsqueda para una actividad en específico y hacia uso de una metodología recursiva para tal fin.

1. ¿Cuál consideran que fue el mayor problema técnico? ¿Qué hicieron para resolverlo?

Compilar y ejecutar el código sin un IDE (BlueJ) escribiéndolo en un note pad y realizando las instrucciones previamente descritas a través de consola.

1. ¿Qué hicieron bien como actividades? ¿Qué se comprometen a hacer para mejorar los resultados?

Analizar detalladamente el camino de ejecución de este laboratorio para especificar correctamente las excepciones y la forman en la que son tratadas.

Continuar dedicando los tiempos apropiados a estas prácticas para así obtener mejores resultados.