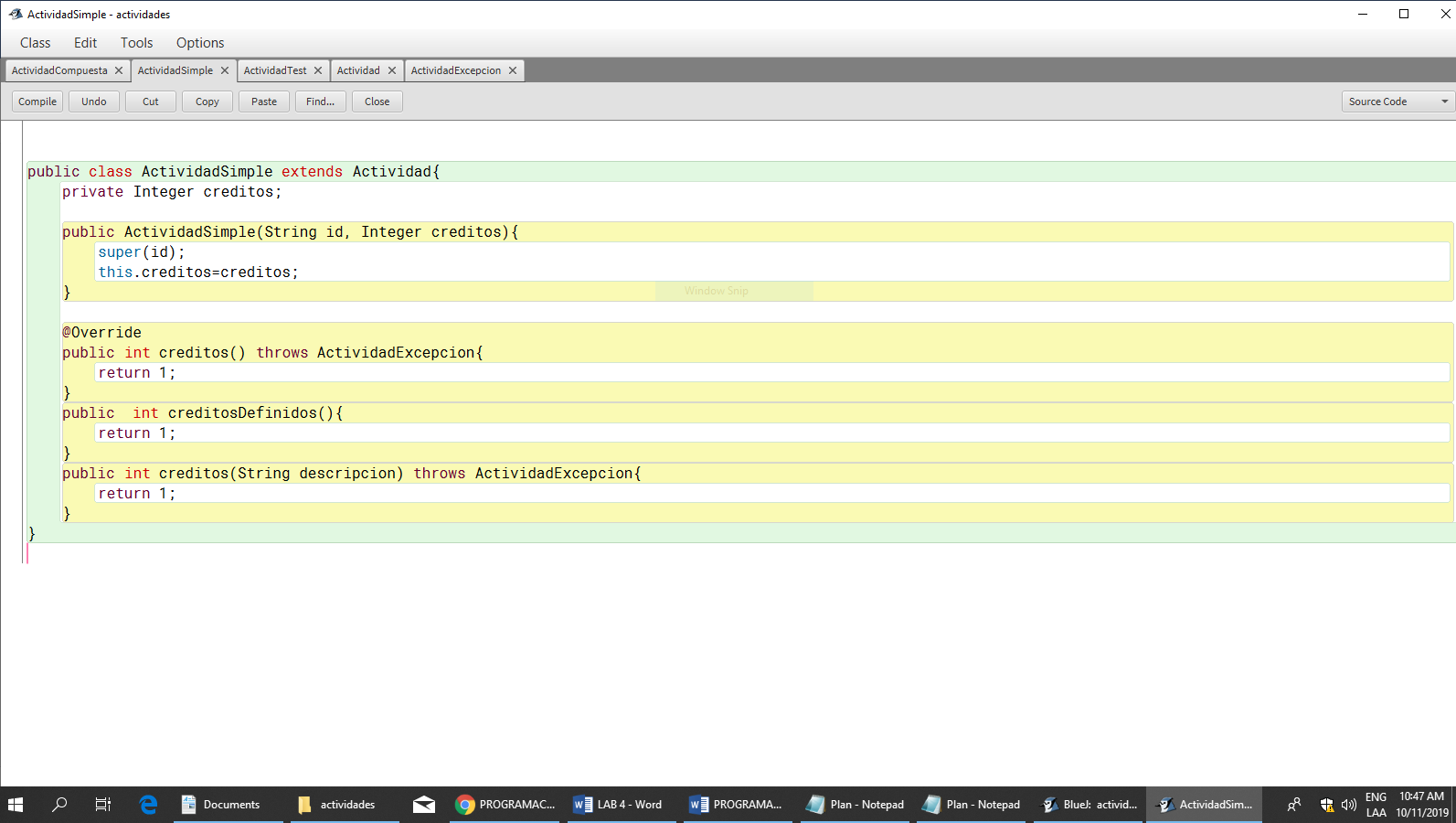
# PROGRAMACIÓN ORIENTADA A OBJETOS Excepciones 2019-02 Laboratorio 4/6 Integrantes: Brayan Burgos y Daniel Alfonso

# EQUIPOS

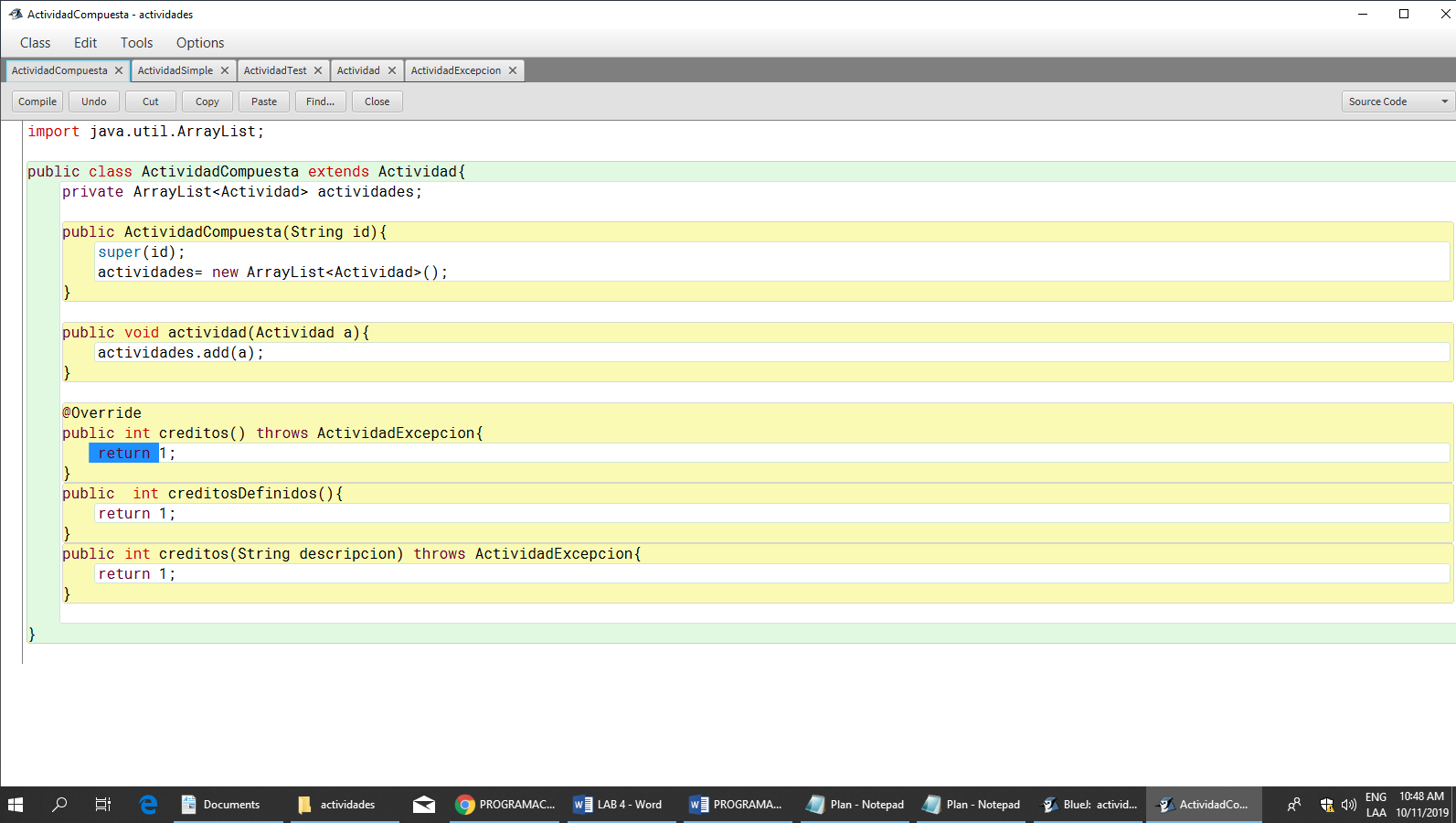
En este punto vamos a aprender a diseñar, codificar y probar usando excepciones. Para esto se van a trabajar algunos métodos de la clase Actividad y la excepción ActividadExcepcion

2. Expliquen por qué el proyecto no compila. Realicen las adiciones necesarias para lograrlo.   
  
No compila ya que las subclases ActividadCompuesta y ActividadSimple no estaban implementando los métodos de la superclase Actividad. Para lograr que compile solo añadimos los métodos abstractos que no están implementados en cada una de las clases.

ActividadSimple:

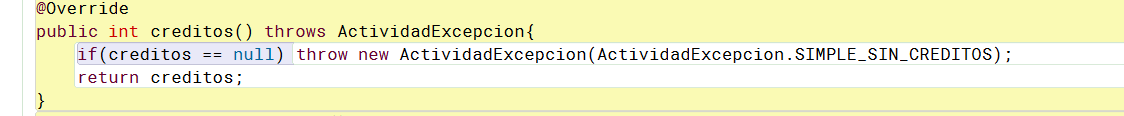


ActividadCompuesta:

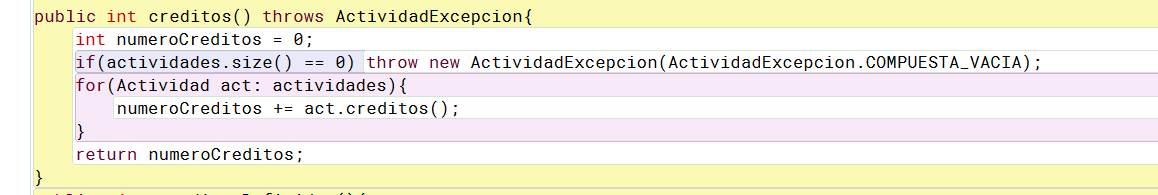


3. Dadas las pruebas, documenten, diseñen y codifiquen el método creditos ().

En actividad simple

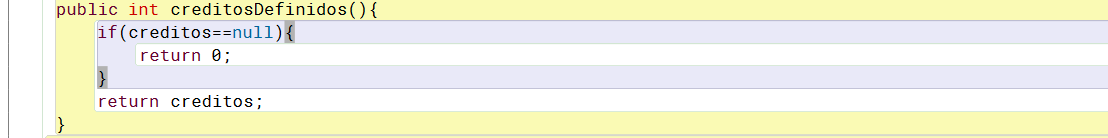


En actividad compuesta

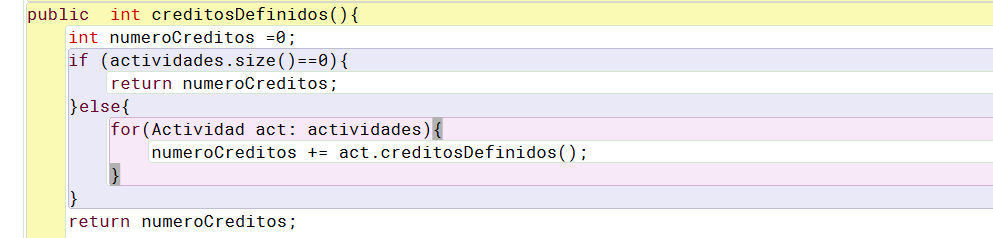


4. Dada la documentación, diseñen, codifiquen y prueben el método creditosDefinidos ().

En actividad simple



En actividad compuesta



5. Documenten, diseñen, codifiquen y prueben el método creditos (String). En este método vamos a incluir dos nuevos casos especiales: 1) no existe una actividad con esa descripción y 2) existen dos actividades con la misma descripción

# EN CONSOLA

1. En su directorio descarguen los archivos contenidos en plan.zip, revisen el contenido.  
   ¿Cuántos archivos se tienen? ¿Cómo están organizados? ¿Cómo deberían estar organizados?  
   Plan contiene los siguientes archivos .java

* Plan
* PlanGUI
* Registro

No están organizados de ninguna manera deberían estar organizados por los paquetes que se dan, por las fuentes .java en una carpeta source y en una carpeta bin los bytecode generados al compilar

Src

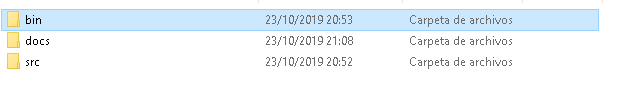
Presentacion

Aplicación

Bin

Docs

1. Prepare los directorios necesarios para ejecutar el proyecto. ¿qué estructura debe tener? ¿qué  
   instrucciones debe dar para ejecutarlo?

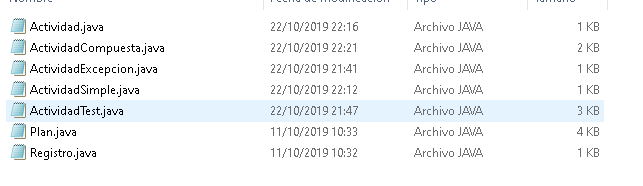


En src

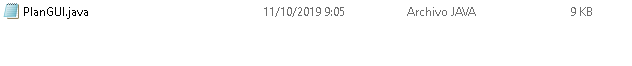


1. Adicione las clases correspondiente a las actividades.

En aplicación



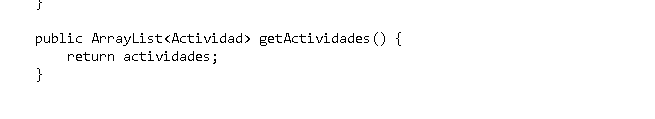
En presentación



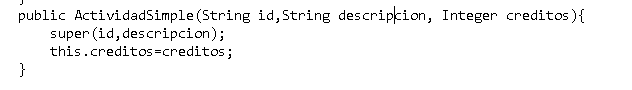
1. Compilen el proyecto, ¿qué métodos tienen que adicionar para lograrlo? (No olviden adicionarlos en el diseño)

Los métodos que debimos adicionar para lograr compilar el proyecto fueron

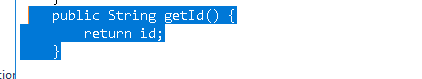
En actividad compuesta



En actividad simple

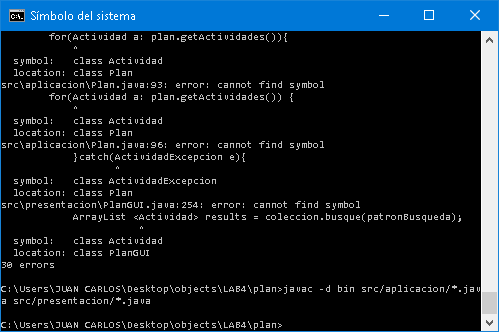


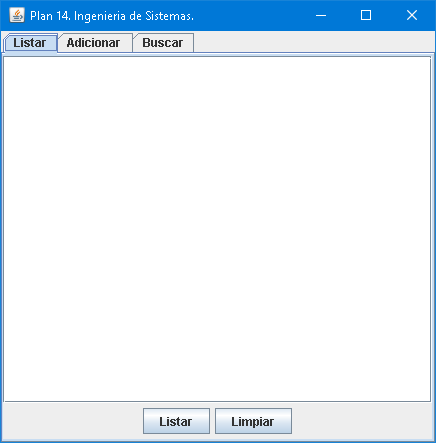
En actividad





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





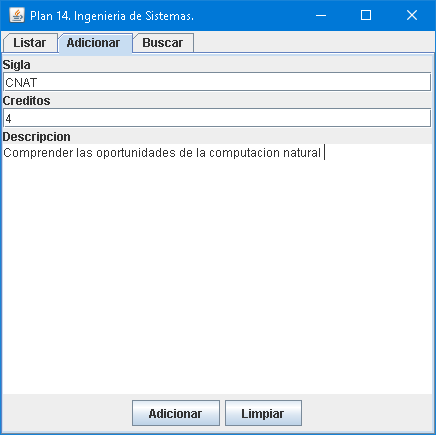
La aplicación permite adiciones materias con sus créditos y descripción, también nos permite listar las materias existentes dentro de un plan. Tambien ofrece la funcionalidad de buscar una materia

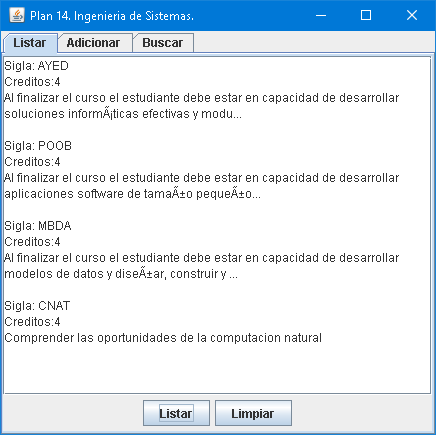
1. Revisen el código del proyecto. ¿De dónde salen la asignatura es iniciales? ¿Qué clase pide  
   que se adicionen? ¿Qué clase los adiciona?

Las asignaturas iniciales se añaden en la clase Plan

# Adicionar y listar. Todo OK.

1. Adicionen una nueva asignatura CNAT 4 créditos Comprender las oportunidades de la computación natural ¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?





1. Revisen el código asociado a adicionar en la capa de presentación y la capa de aplicación. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de aplicación?   
   El responsable en la capa de aplicación es la clase actividad compuesta en el método actividad para y también plan en el método adicione que añade las actividades simples a un “Plan”

El responsable en la capa de presentacion es la clase planGUI con el método accionAdicionar

3. Realicen ingeniería reversa para la capa de aplicación para adicionar. Capturen los resultados de las pruebas de unidad.

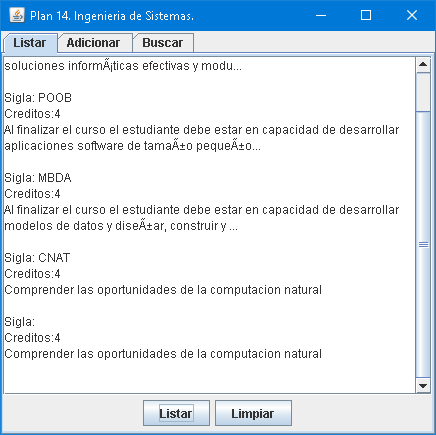
4. Revisen el código asociado a listar en la capa de presentación y la capa de aplicación. ¿Qué método es responsable en la capa de presentación? ¿Qué método en la capa de aplicación?

En la capa de presentación el responsable es el método accionListar y en la capa de aplicación el responsable es la clase plan ya que este cuenta con un método toString para mostrar el arreglo de actividades

5. Realicen ingeniería reversa de la capa de aplicación para listar. Capturen los resultados de las pruebas de unidad.

6. Propongan y ejecuten una prueba de aceptación.

# Adicionar una asignatura. ¿Y si no da la sigla?

1. Adicionen a CNAT sin sigla ¿Qué ocurre? ¿Cómo lo comprueban? Capturen la pantalla. ¿Es adecuado este comportamiento?  
   

Se deja adicionar normalmente una asignatura sin sigla lo comprobamos listando las materias añadidas y se observa.

3. Vamos a evitar la creación de asignaturas con siglas vacías manejando la excepción ActividadExcepcion. Si la asignatura no tiene sigla no la creamos y se lo comunicamos al usuario1. Para esto lo primero que debemos hacer es crear el mensaje en la clase ActividadExcepcion.

4. Analicen el diseño realizado. ¿Qué método debería lanzar la excepción? ¿Qué métodos deberían propagarla? ¿Qué método debería atenderla? Explique claramente.

5. Construya la solución propuesta. Capturen los resultados de las pruebas.

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

# RETROSPECTIVA

1. ¿Cuál fue el tiempo total invertido en el proyecto por cada uno de ustedes? (Horas/Hombre)

El tiempo invertido fueron 12 horas por cada uno

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

El mayor logro para nosotros es, que cada vez vamos entendiendo más lo que es la programación orientada a objetos y logramos aplicar, herencias, superclases, extensibilidad, paquetes y demás

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

El tiempo de entrega a pesar de haber sido extendido, tuvimos bastante problema con el tiempo que pudimos invertir, además del manejo de la sintaxis del lenguaje que aún no es la mejor para avanzar de manera óptima el proceso del proyecto

4. ¿Qué hicieron bien como equipo? ¿Qué se comprometen a hacer para mejorar los resultados?

Como equipo siempre escuchamos la opinión del otro y tratamos de llegar a un acuerdo sobre lo acordado. La utilización de la programación en parejas. Nos comprometemos a esforzarnos y seguir con la disciplina que logramos en este proyecto para lograr los objetivos satisfactoriamente

5. Considerando las prácticas XP del laboratorio. ¿Cuál fue la más útil? ¿Por qué?

Las dos prácticas que conocemos las utilizamos, las iteraciones las utilizamos para dividir en partes más pequeñas los problemas y las más útil fue la programación a pares que ayuda a entenderse y lograr mejores trabajos de calidad en pareja