Escuela Colombiana de Ingeniería Julio Garavito

Andrés Cardozo

Tulio Riaño

Laboratorio 03

Programacion Orientada a Objetos

FECHA:

14/03/2025

**Conociendo**

1. En el directorio descarguen los archivos contenidos en schelling.zip. Revisen el código: a) ¿Cuántos paquetes tiene? d) ¿Cuál es el propósito del paquete presentación? e) ¿Cuál es el propósito del paquete dominio?

Cuando abrimos el documento nos encontramos con dos paquetes: presentacion y domain.

El paquete presentación se encarga de mostrarnos la ciudad.

El paquete dominio nos está implementando el contexto del modelo de Schelling, encontramos la ciudad, donde se colocan items, de los cuales pueden ser personas.

1. Revisen el paquete de dominio, a)¿Cuáles son los diferentes tipos de componentes de este paquete? b) ¿Qué implica cada uno de estos tipos de componentes?

Encontramos 2 clases concretas, una interfaz y una clase abstracta, así como por las relaciones notamos que la clase person implementa la interfaz y extiende la clase abstracta.

Además, podemos ver que persona es de una ciudad, como la ciudad tiene como locations de tipo de la interfaz.

1. Revisen el paquete de presentación, a) ¿Cuántos componentes tiene? b) ¿Cuántos métodos públicos propios (no heredados) ofrece?

Tiene una clase concreta llamada cityGUI , y esta clase tiene la clase CityGUI que extiende JFrame y PhotoCity que extiende JPanel, esto se ve asi ya que PhotoCity es como una clase auxiliar para CityGUI. Los metodos que no heredan son: PhotoCity(CityGUI gui), main(String[] args), gettheCity(), CityGUI()-Constructor, prepareElements(), prepareActions(), ticTacButtonAction().

///todos menos paintComponent(Graphics g)

(con ayuda de referencia 2)

1. Para ejecutar un programa en java, ¿Qué método se debe ejecutar? ¿En qué clase se encuentra?

Se debe ejecutar el método main que se encuentra en la clase CityGUI, ya que es el paquete de presentación.

1. Ejecuten el programa. ¿Qué funcionalidades ofrece? ¿Qué hace actualmente? ¿Por qué? (Deben ejecutar la aplicación java, no crear un objeto como lo veníamos haciendo)

Principalmente al ejecutar el programa se nos abre una pestaña en la que podemos observar una cuadricula en la que se presenta la funcionalidad de tic-tac, esta última no deja hacer nada adicional porque no se ha implementado al 100% el correspondiente método.

**Arquitectura general.**

1. Consulte el significado de las palabras package e import de java.

¿Qué es un paquete? ¿Para qué sirve?

La palabra clave package en Java se utiliza para agrupar clases, interfaces y subpaquetes relacionados. Ayuda a organizar el código y a evitar conflictos de nombres. Los paquetes proporcionan protección de acceso y gestión del espacio de nombres. (tomado de 3)

¿Para qué se importa? Explique su uso en este programa.

La palabra clave import en Java se utiliza para hacer visibles otras clases o paquetes enteros dentro de tu clase actual. Esto te permite utilizar clases de otros paquetes sin tener que hacer referencia a sus nombres completos.

La declaración import se coloca al principio de un archivo fuente Java, después de la declaración del paquete (si existe) y antes de la declaración de la clase. Ayuda a que el código sea más legible y fácil de mantener al permitir el uso de nombres de clase simples en lugar de nombres totalmente cualificados.

En este laboratorio nosotros importamos tanto JFrame como JPanel ya que son clases que se utilizan para crear interfaces gráficas de usuario (GUI):

* JFrame: Crea una ventana principal de la aplicación.
* JPanel: Es un contenedor dentro de la ventana principal para organizar y agrupar otros componentes visuales.

2. Revise el contenido del directorio de trabajo y sus subdirectorios. Describa su

contenido. ¿Qué coincidencia hay entre paquetes y directorios?

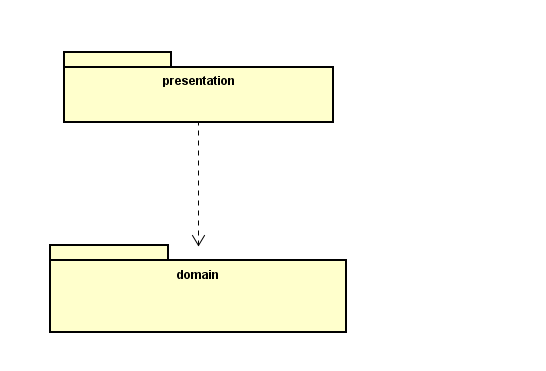
La coincidencia que se presenta entre paquetes y directorios es que este último guarda de la misma manera los paquetes.

El contenido que observamos en el directorio son los respectivos paquetes domain y presentation. En doc se presenta, contenido variado como son varios archivos con extensiones html, algunos java script y carpetas como aplication, domain, presentacion, presentation y resources, cada una de las anteriores tienen archivos que se caracterizan por su extensión html.

3. Adicione al diseño la arquitectura general con un diagrama de paquetes en el que se

presente los paquetes y las relaciones entre ellos. Consulte la referencia en moodle.

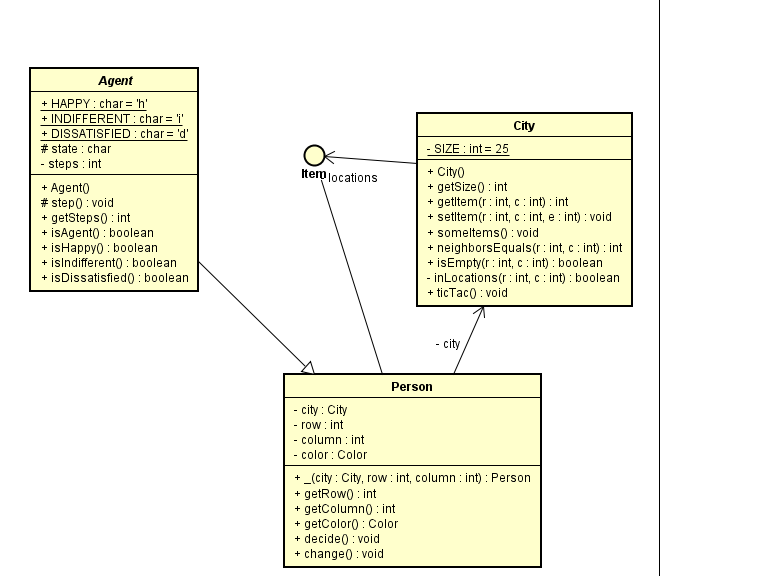
En astah, crear un diagrama de clases (cambiar el nombre por Package Diagram0)



Adicione al diseño la arquitectura general con un diagrama de paquetes en el que se presente los paquetes y las relaciones entre ellos. Consulte la referencia en moodle. En astah, crear un diagrama de clases (cambiar el nombre por Package Diagram0) Arquitectura detallada. [En lab03.doc y schellingasta]

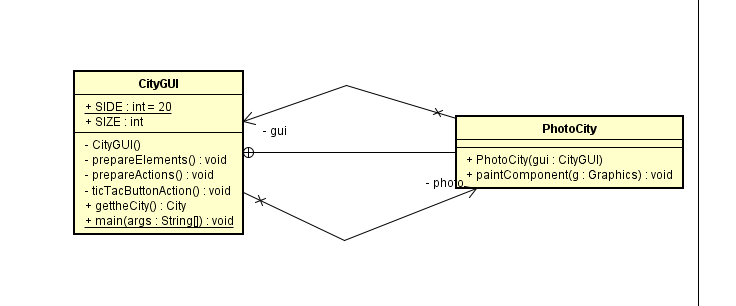
1. Para preparar el poyecto para BDD. Completen el diseño detallado del paquete de dominio. Adicionen el diagrama de clases en el paquete correspondiente. a) ¿Qué componentes hacían falta?

Los componentes que faltaban eran pertenecientes al paquete de dominio y son person y item.



1. Completen el diseño detallado del paquete de presentación. Adicionen el diagrama de clases al paquete correspondiente. a) ¿Por qué hay dos clases y un archivo .java?

Hay dos clases porque una es clase interna (PhotoCity) de la otra (CityGUI). Y un solo archivo Java que es CityGUI.



1. Adicione la clase de pruebas unitarias necesaria para BDD en un paquete independiente de test. (No lo adicione al diagrama de clases) ¿Qué paquete debe usar? ¿Por qué? ¿Asociado a qué clase? ¿Por qué?

El paquete que debe usar test es dominio porque es desde donde se va a realizar en las pruebas en el back, por otra parte, este va a estar asociado a la clase city, ya que la instanciación de esta clase es con el que vamos a interactuar durante las pruebas.

**CICLO 1. Iniciando con las personas normales**

1. Estudie la clase City ¿Qué tipo de colección usa para albergar cosas? ¿Puede recibir personas? ¿Por qué?

La colección que usa para albergar cosas es un vector de 2D, puede recibir personas ya que está implementando la interfaz ítem que se llama locations, por último item es implementado en personas.

1. Estudie el código asociado a la clase Person, ¿en qué estado se crea? ¿qué forma usa para pintarse? ¿cuándo aumenta su tiempo? ¿qué clases definen la clase Person ? Justifique sus respuestas.

Person se crea con un modificador de acceso público, las clases que definen Person son Agent (Clase Abstracta) y Item(interface), se crea en un estado inicial en una fila, columna, ciudad y un color (Blue).

1. Person por ser un Agent, ¿qué atributos tiene? ¿qué puede hacer (métodos)? ¿qué decide hacer distinto? ¿qué no puede hacer distinto a todos los agentes? ¿qué debe aprendar a hacer? Justifique sus respuestas.

Los atributos que hereda persona por ser un Agent son 3 char:

HAPPY='h', INDIFFERENT='i', DISSATISFIED='d'

Y otro Char definido como modificador de acceso protegido llamado state.

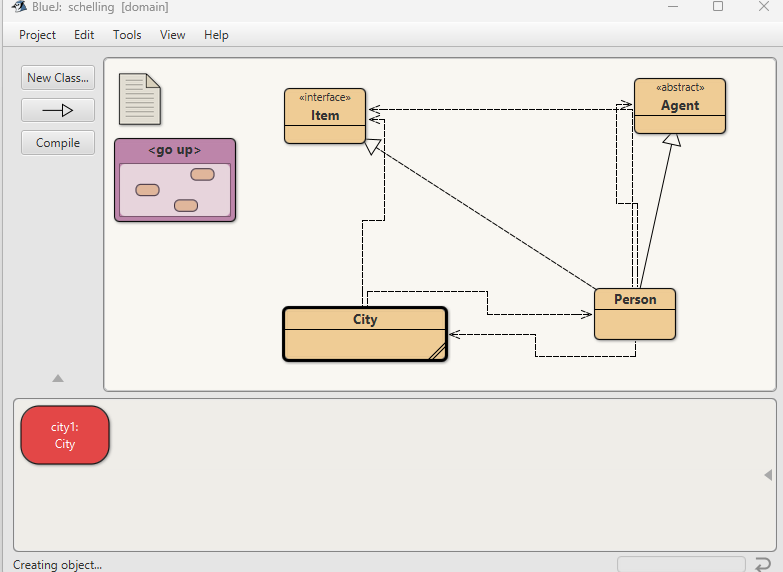


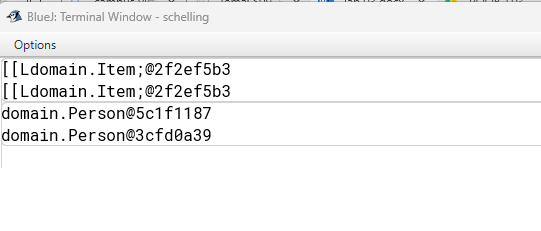
1. Por comportarse como un Item, ¿qué sabe hacer? ¿qué decide hacer distinto? ¿qué no puede hacer distinto? ¿qué debe aprender a hacer? Justifique sus respuestas.

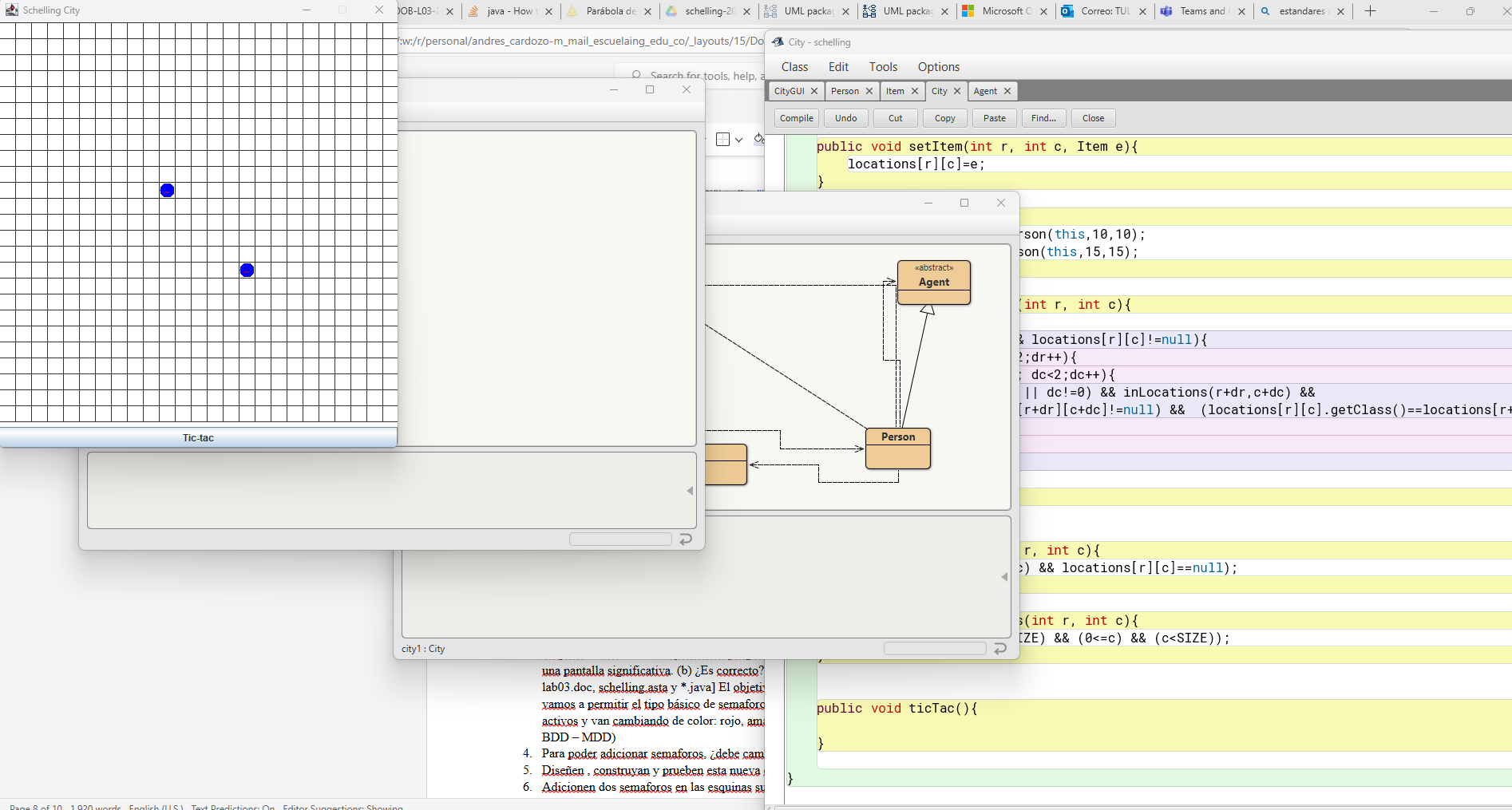
Persona por compartarse como un item sabe retornar el color, isActive, isAgent, shape, change y decide.

1. De acuerdo a lo anterior una Person, ¿Cómo actúa (decide+cambia)?
2. Ahora vamos a crear dos personas en diferentes posiciones (10,10) (15,15) llámelas adan y eva usando el método someItems() . Ejecuten el programa, ¿Qué pasa con las personas? ¿Por qué? Capturen una pantalla significativa.

Se crean las personas y se ubican en la posición correspondiente del tablero, pero visualmente no se observa.

Si imprimimos en el método de setItems podemos ver que si se crean y se asignan en el tablero las personas.



Si usamos el paquete de presentación y ejecutamos se observa en la cuadricula las correspondientes personas.

1. Diseñen, construyan y prueben el método llamado ticTac() de la clase City.
2. ¿Cómo quedarían adan y eva después de uno, dos, cuatro y seis Tic-tac? Ejecuten el programa. Capturen pantallas significativas en momentos correspondientes. ¿Es correcto?

Ciclo 2. Incluyendo a los caminantes [En lab03.doc y schellingasta] (NO OLVIDE BDD – MDD) El objetivo de este punto es permitir recibir personas caminantes . Ellas (i) son rectangulos de color verde; (ii) inician indiferentes; (ii) se mueven hacia el norte1

, (iii) si quedan vecinos

a un item, se ponen felices; (iv) si no logran moverse al sitio que querían, quedan insatisfechos.

1. Para implementar esta nueva persona Walker ¿cuáles métodos se sobre-escriben (overriding)?
2. Diseñen, construyan y prueben esta nueva clase. (Mínimo dos pruebas de unidad)
3. Adicione una pareja de caminantes, llámelas messner y kukuczka, (a) ¿Cómo quedarían después de tres Tic-tac? Ejecuten el programa y hagan tres clics en el botón. Capturen una pantalla significativa. (b) ¿Es correcto? Ciclo 3. Adicionando semaforos [En lab03.doc, schelling.asta y \*.java] El objetivo de este punto es incluir semaforos (sólo vamos a permitir el tipo básico de semaforos) los semaforos son redondos, siempre activos y van cambiando de color: rojo, amarillo, verde, amarillo, rojo, etc (NO OLVIDE BDD – MDD)
4. Para poder adicionar semaforos, ¿debe cambiar en el código de City en algo? ¿por qué?
5. Diseñen , construyan y prueben esta nueva clase. (Mínimo dos pruebas de unidad)
6. Adicionen dos semaforos en las esquinas superiores de la ciudad, llámenlos alarm y alert, (a) ¿Cómo quedarían después de cuatro Tic-tac? Ejecuten el programa y hagan cuatro clics en el botón. Capturen una pantalla significativa. (b) ¿Es correcto?

Ciclo 4. Nueva persona: Proponiendo y diseñando El objetivo de este punto es permitir recibir en un nuevo tipo de Item. (NO OLVIDE BDD – MDD)

1. Propongan, describan e implementen el nuevo tipo de persona. (Mínimo dos pruebas de unidad)
2. Considerando una pareja de ellas con el apellido de ustedes. (a) Piensen en otra prueba significativa y expliquen la intención. (b) Codifiquen la prueba de unidad correspondiente y capturen la pantalla de resultados de ejecución de la prueba. (c) Ejecuten el programa con esa prueba como prueba de aceptación y capturen las pantallas correspondientes. Ciclo 5. Nuevo item: Proponiendo y diseñando El objetivo de este punto es permitir recibir un nuevo item (no persona) en la ciudad (NO OLVIDE BDD – MDD)
3. Propongan, describan e implementen el nuevo tipo de item. (Mínimo dos pruebas de unidad)
4. Considerando un par de ellos con el nombre de ustedes. (a) Piensen en otra prueba significativa y expliquen la intención. (b) Codifiquen la prueba de unidad correspondiente y capturen la pantalla de resultados de ejecución de la prueba. (c) Ejecuten el programa con esa prueba como prueba de aceptación y capturen las pantallas correspondientes.

1 Primero, todos los caminantes deciden hacia donde se van a mover y luego todos los caminantes se mueven. Dos caminantes no se pueden mover hacia la misma posición.

Ciclo 6. BONO. Persona Schelling [[http://ncase.me/polygons-es/]](http://ncase.me/polygons-es/%5D) La persona Schelling se rigue por las siguientes reglas:  Las personas Schelling sólo se mueven si están insatisfechos con su vecindad.  Cuando están bien donde están, las personas Schelling no se mueven.  Están indiferentes si todos los vecinos son como ellos o no tienen vecinos.  Están insatisfechos si menos de 1⁄3 de los vecinos no son como ellos.  Están satisfechos si más de 1⁄3 de los vecinos son como ellos y no todos son como ellos. Primero todas las personas Schelling toman la decisión de lo que pasará en el tiempo siguiente y luego la realizan.

Empaquetando la versión final para el usuario. [En lab03.doc, schelling.asta , \*.java, schelling.jar]

1. Revise las opciones de BlueJ para empaquetar su programa entregable en un archivo .jar. Genere el archivo correspondiente.
2. Consulte el comando java para ejecutar un archivo jar. ejecutennlo ¿qué pasa?
3. ¿Qué ventajas tiene esta forma de entregar los proyectos? Explique claramente.

Referencias

1. <https://dar10comyr.blogspot.com/2014/10/jframe.html>
2. <https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/javax/swing/JFrame.html>
3. <https://www.datacamp.com/es/doc/java/package>
4. <https://www.datacamp.com/es/doc/java/import>