LAB03 PROGRAMACIÓN ORIENTADA A OBJETOS

Herencia e interfaces

ADEMAS Java desde consola

PRESENTADO A:

MARIA IRMA DIAZ ROSO

PRESENTADO POR:

CAROLINA CEPEDA

ESCUELA COLOMBIANA DE INGENIERIA JULIO GARAVITO

BOGOTA D.C

2025-1

***CONOCIENDO***

*1. En el directorio descarguen los archivos contenidos en schelling.zip. Revisen el*

*código:*

a) ¿Cuántos paquetes tiene?

Se tienden dos paquetes presentation y domain

b) ¿Cuál es el propósito del paquete presentación?

El propósito del paquete presentación es agrupar las clases que tienen una característica en común, en este caso, la parte gráfica del proyecto.

c) ¿Cuál es el propósito del paquete dominio?

Su propósito es agrupar las clases que se encargan de los aspectos detrás de los gráficos que se pueden observar, es decir, la comparación de sus ubicaciones y otras funcionalidades del proyecto.

*2. Revisen el paquete de dominio,*

a) ¿Cuáles son los diferentes tipos de componentes de

este paquete?

Este paquete tiene una interfaz llamada item, una clase abstracta agent y dos clases concretas llamadas person y city.

b) ¿Qué implica cada uno de estos tipos de componentes?

La interfaz define un conjunto de métodos que deben ser implementados por las clases que la usen, por lo que funciona como un contrato de comportamiento. Por otro lado, una clase abstracta actúa como una plantilla para otras clases, permitiendo definir métodos y atributos base, pero no pueden ser instanciadas. Por último, las clases concretas sí pueden ser instanciadas y usadas de forma directa en un programa.

*3. Revisen el paquete de presentación,*

*a) ¿Cuántos componentes tiene?*

Tiene dos componentes, una clase llamada CityGUI y una referencia a la pantalla principal

*b) ¿Cuántos métodos públicos propios (no heredados) ofrece?*

Ofrece 2 métodos públicos; main y PhotoCity.

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

*encuentra?*

Se debe ejecutar el método main de la clase CityGUI.

*5. Ejecuten el programa. ¿Qué funcionalidades ofrece? ¿Qué hace actualmente? ¿Por*

*qué?*

El programa ofrece un botón llamado Tic-tac el cual todavía no está en funcionamiento debido a que este está basado en un método de la clase City que todavía no está definido.

Imagen en blanco y negro

El contenido generado por IA puede ser incorrecto.

***ARQUITECTURA GENERAL***

*1. Consulte el significado de las palabras package e import de java. ¿Qué es un*

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

Un paquete es una unidad de almacenamiento en donde se agrupan varias clases e interfaces, hacer estas agrupaciones es muy útil para dividir un programa en partes más pequeñas y fáciles de manejar. A su vez, se importan paquetes para poder usar las herramientas que nos sean útiles de estos en otras áreas del trabajo.

En este programa, se hace uso de import para importar los siguientes paquetes en la clase CityGUI , un ejemplo de estos paquetes importados es el paquete domain que es, como se dijo anteriormente, el que lleva la creación de objetos con sus atributos , por lo que de esta forma puede usarse la información que se almacena y crea allí de manera inmediata en la parte gráfica de la aplicación.

Texto, Carta

El contenido generado por IA puede ser incorrecto.

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

*contenido. ¿Qué coincidencia hay entre paquetes y directorios?*

En los paquetes y directorios encontramos que ambos contienen las clases definidas dentro de ellos.

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

*presente los paquetes y las relaciones entre ellos.*

Diagrama

El contenido generado por IA puede ser incorrecto.

***ARQUITECTURA DETALLADA***

*1. Para preparar el proyecto para BDD. Completen el diseño detallado del paquete de*

*dominio. Adicionen el diagrama de clases en el paquete correspondiente.*

*a) ¿Qué componentes hacían falta?*

Hacían falta los componentes de la clase Person y la interfaz item.

*2. 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 y un archivo .java debido a que en el paquete se hace uso de librerías propias de java con el fin de manejar el aspecto gráfico del proyecto.

*3. Adicione la clase de pruebas unitarias necesaria para BDD en un paquete*

*independiente de prueba. (No lo adicione al diagrama de clases) ¿Qué paquete debe*

*usar? ¿Por qué? ¿Asociado a qué clase? ¿Por qué?*

Debe usarse un paquete independiente debido a que no hace parte de la presentación del producto final (presentation) ni del paquete de la lógica y uso de objetos abstractos (domain), sino, más bien, que es una parte del proceso de creación del software planteado para ser usado por el programador. A su vez, este paquete debería estar asociado a la clase City, ya que está conectada a todos los aspectos del paquete domain y los posibles errores de lógica o casos de prueba no planteados.

Diagrama

El contenido generado por IA puede ser incorrecto.

***Ciclo 1. Iniciando con los personas normales***

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

*personas? ¿Por qué?*

La clase de colección que usa la clase City es un arreglo de dos dimensiones compuesto por ítems, debido a que la clase Person implementa la interfaz Item, los objetos de tipo Person pueden ser almacenados dentro de el arreglo. Esto es posible por el principio de polimorfismo, el cual permite que una referencia de tipo Item pueda apuntar a cualquier objeto de cualquier clase que implemente esta interfaz.

*2. 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.*

Una persona se crea en estado “dissatisfied” debido a que ese es el estado predeterminado en el que se crea un agente, clase de la cual hereda sus características. A su vez, usa la forma ROUND para pintarse, ya que esa es la forma predeterminada de la interfaz implementada ( Item). A su vez, la clase agente es usada para definir a la clase Person debido a que es la clase de la que hereda sus comportamientos.

*3. 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*

*aprender a hacer? Justifique sus respuestas.*

Por ser un Agent, Person tiene los atributos state, steps y unos atributos tipo char que no pueden ser modificados que son las definiciones de los posibles estados que puede tener un objeto Agent. Debido a la herencia de métodos, Person puede aumentar su cantidad de pasos y devolverla, verificar su estado de ánimo, saber si está feliz, indiferente o insatisfecho.

Person decide decidir su estado de ánimo de forma diferente mediante un cálculo relacionado a su cantidad de pasos, también decide establecer su color cuando se construye de forma predeterminada azul, usar un sistema de ubicación mediante filas y columnas, y ser parte de una ciudad.

*4. 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.*

Por comportarse como un item, tiene una forma que está representada con un 1

*5. De acuerdo con lo anterior una Person, ¿Cómo actúa (decide+cambia)?*

La persona actúa respecto a su cantidad de pasos, cuando decide en que estado de ánimo está hace un cálculo respecto a estos y cuando cambia aumenta su número de pasos en 1, sabe retornar que su color es negro y sabe verificar que está activo. La clase decide plantear su propia forma de decidir y cambiar su estado de ánimo, pero no puede decidir qué color devuelve y de qué forma es.

*6. 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*

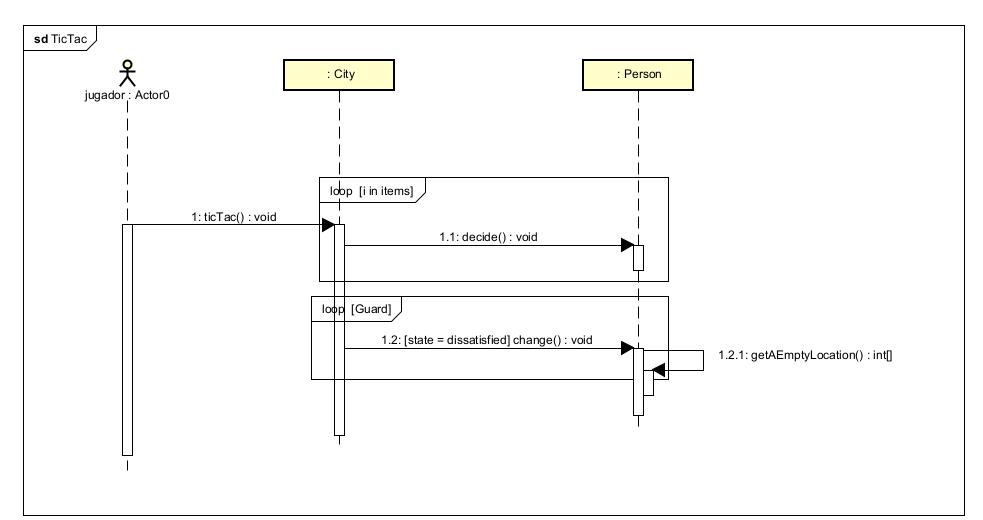
*las personas? ¿Por qué? Capturen una pantalla significativa.*

Las personas son mostradas en la ciudad ( cuadricula) y ambas son representadas como círculos, a su vez, están insatisfechos y no tienen pasos disponibles.

*Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.*

*7. Diseñen, construyan y prueben el método llamado ticTac() de la clase City.*



***Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.***

8. ¿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?

Adan y Eva deberían estar acercándose entre ellos, entre mas ticks mas cerca están ya que dan un paso en dirección del otro

1 tick

Gráfico

El contenido generado por IA puede ser incorrecto.

2 ticks

Imagen que contiene biombo, edificio

El contenido generado por IA puede ser incorrecto.

6 ticks

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

A pesar de que el programa acerca a las personas no es totalmente correcto ya que debería ser más eficiente a la hora de acercarse entre ellas.

***Ciclo 2. Incluyendo a los caminantes***

*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)?

* Se sobreescribe el método shape
* Se sobre escribe el constructor
* Se sobreescribe el método decide
* Se sobreescribe el método change

2. Diseñen, construyan y prueben esta nueva clase. (Mínimo dos pruebas de unidad)

3. Adicione una pareja de camin

antes, 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.

Inicial:

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

Después de tres tic-Tac:

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

(b) ¿Es correcto?

Es correcto.

Pruebas de unidad

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

***Ciclo 3. Adicionando semáforos***

*El objetivo de este punto es incluir semáforos (sólo vamos a permitir el tipo básico de*

*semáforos) los semáforos son redondos, siempre activos y van cambiando de color: rojo,*

*amarillo, verde, amarillo, rojo, etc.*

1. Para poder adicionar semáforos, ¿debe cambiar en el código de City en algo? ¿por

qué?

No. Además de añadir en someitems no es necesario cambiar nada en City.

2. Diseñen, construyan y prueben esta nueva clase. (Mínimo dos pruebas de unidad)

Diagrama

El contenido generado por IA puede ser incorrecto.





3. 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.

Inicial

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

4 clics después

(b) ¿Es correcto?

No, debido a que no cambian del color amarillo.

***Ciclo 4. Nueva persona: Proponiendo y diseñando***

El objetivo de este punto es permitir recibir en un nuevo tipo de Item.

1. Propongan, describan e implementen el nuevo tipo de persona.
2. *Personas solitarias: Ellas son círculos de color rosado; inician diferentes, se enfocan en la cantidad de vecinos que tengan:*
   * *Si tienen más de 3 vecinos solitarios están insatisfechas*
   * *Si tienen exactamente 3 vecinos solitarios son indiferentes*
   * *Si tienen menos de 3 vecinos solitarios están felices.*

*Pruebas*

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

2. Considerando una pareja de ellas con el apellido de ustedes.

(a) Piensen en otra prueba significativa y expliquen la intención.

Se hace la prueba cuando dos personas Solitarias están rodeadas de dos personas solitarias más y deben estar indiferentes.

(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. Como se puede observar, las personas Solitarias son indiferentes.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

Al dar clic al botón Tic-Tac deciden moverse

Imagen que contiene biombo, juego, edificio, reloj

El contenido generado por IA puede ser incorrecto.

Y al volver dar clic al botón, revisan su entorno y cambian su estado a felices.

Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.

***Ciclo 5. Nuevo item: Proponiendo y diseñando***

El objetivo de este punto es permitir recibir un nuevo item (no persona) en la ciudad

1. Propongan, describan e implementen el nuevo tipo de item.



Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

2. Considerando un par de ellos con el nombre de ustedes.

(a) Piensen en otra prueba significativa y expliquen la intención.

Un hoyo debe ser un obstáculo para otros objetos, por ejemplo, un caminante que siempre va hacia delante.Por lo que se revisa si cumple con esta condici+on.

(b) Codifiquen la prueba de unidad correspondiente y capturen la pantalla de resultados de ejecución de la prueba.



(c) Ejecuten el programa con esa.

Inicialmente:

Después de tic-tac:

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.

***DE BLUEJ A CONSOLA***

En esta sección del laboratorio vamos a aprender a usar java desde consola. Para esto se va a trabajar con el proyecto del punto anterior.

***Comandos básicos del sistema operativo***

Antes de iniciar debemos repasar los comandos básicos del manejo de la consola.

1. Investiguen los comandos para moverse en la estructura de directorios: crear, borrar, listar su contenido y copiar o eliminar un archivo.

2. Organicen un nuevo directorio con la estructura propuesta para probar desde allí su habilidad con los comandos de consola. Consulten y capturen el contenido de su directorio

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.3. En el directorio copien únicamente los archivos \*.java del paquete de aplicación. Consulte y capture el contenido de src/domain

***Estructura de proyectos java***

En java los proyectos se estructuran considerando tres directorios básicos.

Tabla

El contenido generado por IA puede ser incorrecto.

1. Investiguen los archivos que deben quedar en cada una de esas carpetas y la organización interna de cada una de ellas.

2. ¿Qué archivos debería copiar del proyecto original al directorio bin? ¿Por qué? Cópielos y consulte y capture el contenido del directorio que modificó.

***Comandos de java***

1. Consulte para qué sirven cada uno de los siguientes comandos:

javac

java

javadoc

jar

2. Cree una sesión de consola y consulte en línea las opciones de los comandos java y javac. Capture las pantallas.

3. Busque la opción que sirve para conocer la versión a que corresponden estos dos comandos. Documente el resultado.

***Compilando***

1. Utilizando el comando javac, desde el directorio raiz (desde schelling con una sóla instrucción), compile el proyecto. ¿Qué instrucción completa tuvo que dar a la consola para compilar TODO el proyecto? Tenga presente que se pide un único comando y que los archivos compilados deben quedar en los directorios respectivos.

2. Revise de nuevo el contenido del directorio de trabajo y sus subdirectorios. ¿Cuáles nuevos archivos aparecen ahora y dónde se ubican?

***Documentando***

1. Utilizando el comando javadoc, desde el directorio raiz, genere la documentación (API) en formato html, en este directorio. ¿cuál es el comando completo para generar esta documentación?

2. ¿Cuál archivo hay qué abrir para empezar a navegar por la documentación? Ábralo y capture la pantalla.

***Ejecutando***

4. Empleando el comando java, desde el directorio raiz, ejecute el programa. ¿Cómo utilizó este comando?

***Probando***

1. Adicione ahora los archivos del directorio pruebas y trate de compilar nuevamente el programa.Tenga en cuenta que estas clases requieren la librería junit 4.8. ¿Cómo se incluye un paquete para compilar? ¿Qué instrucción completa tuvo que dar a la consola para compilar?

2. Ejecute desde consola las pruebas . ¿Cómo utilizó este comando?. Puede ver ejemplos de cómo ejecutar el“test runner”en How to run JUnit test cases from the command line 3. Pegue en su documento el resultado de las pruebas

***Empaquetando***

1. Consulte como utilizar desde consola el comando jar para empaquetar su programa entregable en un archivo .jar, que contenga los archivos bytecode necesarios (no las fuentes ni las clases de prueba), y que se pueda ejecutar al instalarlo en cualquier directorio, con solo tener la máquina virtual de java y su entorno de ejecución (JRE). ¿Cómo empaquetó jar ?

2. ¿Cómo se ejecuta el proyecto empaquetado?

***RETROSPECTIVA***

1. ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes?

20 horas 21/03

2. ¿Cuál es el estado actual de laboratorio? ¿Por qué? (Para cada método incluya su estado)

3. Considerando las prácticas XP del laboratorio de hoy ¿por qué consideran que son importantes?

4. ¿Cuál consideran fue su mayor logro? ¿Por qué? ¿Cuál consideran que fue su mayor problema? ¿Qué hicieron para resolverlo?

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

6. ¿Qué referencias usaron? ¿Cuál fue la más útil? Incluyan citas con estándares adecuados.

***REFERENCIAS***

Charte, F. (2021, 9 febrero). Paquetes en Java: qué son, para qué se utilizan, y cómo se usan (con vídeo). *campusMVP.es*. <https://www.campusmvp.es/recursos/post/paquetes-en-java-que-son-para-que-se-utilizan-y-como-se-usan.aspx>