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 esta 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). Además, aumenta su tiempo cuando \_\_\_\_\_\_\_\_\_\_. 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.*

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?

***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

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***

*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*

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

qué?

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

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. (b) ¿Es correcto?

***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>