# LABORATORIO #3: HERENCIA E INTERFACES

## CONOCIENDO

1. Revisando el código de la aplicación
2. ¿Cuántos paquetes tiene?

Tiene dos paquetes: presentation, domain

1. ¿Cuál es el propósito del paquete presentación?

Crear y diseñar la simulación del autómata celular.

“Belleza”

1. ¿Cuál es el propósito del paquete dominio?

Se encarga de la parte lógica para dar solución al problema. “Inteligencia”

1. Revidando el paquete dominio
2. ¿Cuáles son los diferentes tipos de componentes de este paquete?

Aparecen dos tipos de componentes: Interface y Abstract

1. ¿Qué implica cada uno de estos tipos de componentes?

Para el caso de Interface permite definir la funcionalidad, no implementarla.

Por otro lado, que sea Abstract permite crear funcionalidades que las subclases pueden implementar o anular.

1. Revisando el paquete presentación
2. ¿Cuántos componentes tiene?

No tiene, 0

1. ¿Cuántos métodos ofrece?

Ofrece dos métodos: getAutomata(), main(java.lang.String[ ] args)

1. Para ejecutar un programa en java
2. ¿Qué método se debe ejecutar?

Se debe ejecutar el método void main(String[ ] args)

1. ¿En qué paquete de autómata se encuentra? ¿En qué clase?

En el paquete de presentation, en la clase AutomataGUI

1. ¿Por qué?

Este paquete se encarga de simular del problema.

1. Ejecutando el programa
2. ¿Qué funcionalidades ofrece?

Solamente ofrece el botón de Tic-tac.

1. ¿Qué hace actualmente?

En este momento solo tiene la representación visual pero no tiene ninguna función.

## ARQUITECTURA GENERAL

1. Consulte el significado de las palabras package e import de java
2. ¿Para qué sirve package?

Es un contenedor de clases que permite agrupar las distintas partes de un programa y que este tiene una funcionalidad, definiendo la ubicación de dichas clases en un directorio.

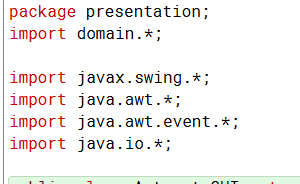
1. ¿Para qué sirve import?

El import en Java sirve para hacer uso de las librerías propias de Java, se usa la palabra clave ***import*** seguido de la ruta del paquete o clase que se desea agregar al proyecto.

1. Explique su uso en esta aplicación.

En esta aplicación permiten empaquetar clases e importar librerías de otros paquetes.

Como ejemplo:



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

Se presentan los paquetes de la aplicación (presentation, domain, doc) y en la parte de los subdirectorios tenemos las clases que corresponden a cada paquete.

En el paquete de presentation está la clase AutomataGUI y la PhotoAutomata, por otro lado, en el paquete domain están las clases Item, Agent, Cell, CellularAutomata.

1. ¿Qué coincidencia hay entre paquetes y directorios?

Ambos permiten agrupar información ordenada.

1. 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.



## 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.
2. ¿Qué componentes hacían falta?

Faltaban las clases Item, Cell

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

Hay dos clases porque …….

1. Adicione en las fuentes la clase de pruebas unitarias necesaria para BDD. ¿En qué paquete debe estar? ¿Por qué? ¿Asociado a qué clase? ¿Por qué?

Se adiciona la clase UnitTest al paquete de domain ya que este tiene contiene la lógica de la solución al problema, se asocia a la clase CellularAutomata ya que es la principal en este paquete.

## CICLO1: INDICANDO CON LAS CÉLULAS NORMALES

1. Estudie la clase CellularAutomata ¿Qué tipo de colección usa para albergar los items?

¿Puede recibir células? ¿Por qué?

La clase CellularAutomata usa la colección de Array de dos dimensiones para albergar los ítems

Esta clase puede recibir células debido a que Item es una clase hija de Cell

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

La célula se crea en estado alive dado que al instanciar Cell este es su estado inicial.

La forma que usa para pintarse es ROUND ya que esta es definida por la clase Item.

Aumenta la edad cuando se transforma (turn()) esto está definido en la clase Agent.

Las clases que definen Cell son Item y Agent ya que estas son clases padre de la clase Cell, es decir tiene atributos y métodos que heredan.

1. Cell por ser un Agent, ¿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.

Los atributos heredados son UNKNOWN, ALIVE, DEAD, state, age. Tiene los métodos: turn(), getAge(), isAlive()

La clase Cell decide hacer distinto: decide(), getColor(), getColumn(), getRow(), change()

La clase Cell no puede hacer distinto a los agentes es transformarse, turn(), lo que hará que su edad aumente

Debe aprender a dar su ubicación. Ya que la clase Agent no le hereda el comportamiento para conocer su ubicación y este debe adquirirlo.

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.

Por comportarse como un Item va a saber si la célula esta viva o no, su color, su forma, decide adicionar los métodos getColumn(), getRow(), es decir, dar la ubicación de la célula.

Debe aprender a dar su ubicación y a cambiar de estado, ya que la clase Item no le define cual será ese comportamiento

1. Ahora vamos a crear dos células en diferentes posiciones (1,1) (2,2) llámelos indiana y 007 usando el método someItems() . Ejecuten el programa, ¿Qué pasa con las células? ¿Por qué? Capturen una pantalla significativa.

Texto

Descripción generada automáticamente con confianza media

Imagen que contiene Gráfico

Descripción generada automáticamente

Las células se ubican en la posición indicada y son muertas.

1. Diseñen, construyan y prueben el método llamado ticTac() de la clase

CellularAutomata.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. ¿Cómo quedarían indiana y 007 después de uno, dos y tres Tic-tac? Ejecuten el programa y hagan tres clics en el botón. Capturen una pantalla significativa. ¿Es correcto?

1-Clic

Gráfico

Descripción generada automáticamente con confianza baja

3-Clics

Gráfico

Descripción generada automáticamente con confianza baja

2-Clics

Gráfico

Descripción generada automáticamente con confianza baja

Después de hacer tres clics en el botón indiana y 007 deberían morir, sin embargo, esto no lo hacen, por lo tanto, no es correcto.

Porque primero toman una decisión y luego cambian.

## CICLO 2: INCLUYENDO A LAS CÉLULAS INQUIETAS

## CICLO 3: ADICIONANDO BOMBILLOS

## CICLO 4: NUEVA CÉLULA, PROPONIENDO Y DISEÑANDO

## CICLO 5: NUEVO ITEM, PROPONIENDO Y DISEÑANDO

## Retrospectiva

### ¿Cuál fue el tiempo total invertido en el laboratorio por cada uno de ustedes? (Horas/Nombre)

(10/Angie Natalia Mojica)  
(10/Daniel Antonio Santanilla)

### ¿Cuál es el estado actual del laboratorio? ¿por qué?

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

### ¿Cuál consideran fue el mayor logro? ¿Por qué?

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

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