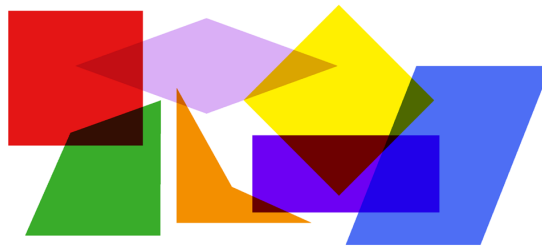


PRIMERA PRÀCTICA DE PROGRAMACIÓ

Representar la informació per a crear imatges a partir de polígons

Introducció al problema

Com es guarda la informació per poder aconseguir imatges d'aquest estil?



El contingut d'aquesta imatge és un conjunt de polígons. Per poder pintar cada polígon de la imatge hem de tenir guardades algunes dades, el color i, sobretot, la informació geomètrica que el defineix i que és el conjunt ordenat de vèrtexs.

El propòsit d'aquesta primera pràctica de programació és dissenyar en JAVA les classes que guardin la informació geomètrica de cada polígon. Treballarem la pràctica en tres sessions de laboratori i anirem construint-la per parts. A les sessions de laboratori no només es treballarà la pràctica, primer explicarem i validarem els conceptes que es necessiten sobre altres exemples i llavors els aplicarem a la part de la pràctica.

Instruccions generals pel desenvolupament de l'exercici

La primera pràctica es realitzarà en tres parts. Cada estudiant entregarà el codi de cada una de les parts en les dates previstes i avaluarà codis de companys/es en els dies posteriors al lliurament. El lliurament i avaluació es farà a través del Campus Virtual (moodle). Els detalls de dates s'indiquen en el següent apartat.

L'avaluació de pràctiques de companys/es és una eina més en el procés d'aprenentatge, serveix per entendre altres formes de resoldre el problema i analitzar la correctesa ajudats per la rúbrica amb una solució correcta.

La nota que rebràs d'aquesta pràctica estarà formada per les notes rebudes en les avaluacions que et facin els companys/es, una nota calculada a partir de la correctesa de les avaluacions que has fet tu i la nota d'un qüestionari final d'autoavaluació. Pots adreçar-te al professorat de laboratori per dubtes o disconformitats en les avaluacions rebudes.

Dates i planificació

Part 1 de la pràctica. A la sessió 2 de laboratori es presenten i treballen conceptes per ajudar a resoldre la primera part de la pràctica i es deixa temps per preparar-la i resoldre dubtes.

Dates de tramesa del resultat de la part 1: des del dia 3/10/23 al dimarts 10/10/23 a les **16:00 hores**.

El dimarts dia 10/10 a les 17:00 hores es mostra la solució d'aquesta part en una rúbrica amb criteris de valoració. És interessant aquesta solució perquè serveix com punt de partida de les altres dues parts.

Avaluació: des del dia 11/10/23 al 16/10/23 (quan no s'especifica hora és la primera hora en el dia d'obertura i l'última del dia per al tancament).

Part 2 de la pràctica. Es presenten conceptes i es treballa a la sessió 3.

Dates de tramesa del resultat de la part 2: des del dia 10/10/23 al dimarts 22/10/23.

El 23/10 es mostra la solució d'aquesta part en una rúbrica amb criteris de valoració.

Avaluació: des del dia 23/10/23 al 27/10/23.

Part 3 de la pràctica. Es presenten conceptes i es treballa a la sessió 4.

Dates de tramesa del resultat de la part 3: des del dia 17/10/23 al dimarts 28/10/23.

El 29/10 es mostra la solució d'aquesta part en una rúbrica amb criteris de valoració.

Avaluació: des del dia 29/10/23 al 2/11/23.

Qüestionari final d'autoavaluació de l'aprenentatge en la primera pràctica:

divendres 03/11/23, franja on està actiu el qüestionari de 12:00 a 18:00

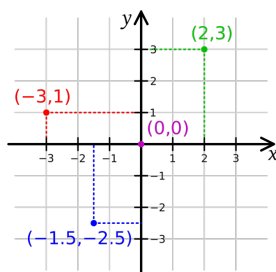
el qüestionari és d'un sol intent i durada limitada

Enunciat i tasques a realitzar en cada part

Part 1. L'element Punt de dos dimensions.

L'element més simple a guardar és el punt.

Ens limitarem a treballar amb punts de dos dimensions, definits per les seves coordenades (x, y) en un sistema geomètric euclidià clàssic. Podem treballar també en aritmètica entera.



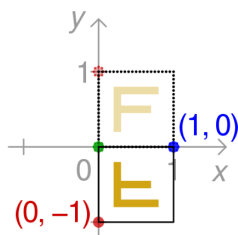
La primera part de la pràctica 1 consistirà en crear la classe Punt amb el conjunt d'operacions que s'indiquen a continuació i validar aquestes operacions en una classe aplicació, UsaPunt.

La classe Punt ha d'estar formada per:

- Els atributs per guardar la informació d'un punt de dues dimensions (aritmètica entera)
- Mètodes:
 - Constructors
 - Getters/setters
 - toString
 - crear un duplicat del punt
 - Comprovar si un punt és igual a un altre
 - Calcular la distància entre un punt i un altre utilitzant la següent fórmula:

$$dist = \sqrt{(x - x')^2 + (y - y')^2}$$

- Calcular la reflexió d'un punt sobre l'eix de les x. L'operació es mostra a la següent imatge i podeu comprovar que el punt resultant de la reflexió manté la mateixa component x i li canvia el signe de la component y. Exemple: el punt resultant de calcular la reflexió sobre (3, 4) és el (3, -4)

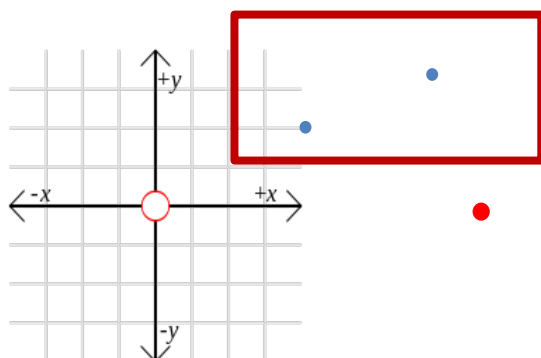


La classe UsaPunt ha de tenir la classe que conté el main i que validi mitjançant un joc de proves complert tots els mètodes de la classe Punt.

Part 2. L'element Punt restringit a una certa regió de l'espai 2D.

Ja disposem de la classe que permet representar l'element més simple, el punt (i es podrà consultar una possible solució en la part d'avaluació del taller de la primera part de la pràctica).

Ara volem que els objectes Punt que creem estiguin sempre dins una regió rectangular definida pels seus extrems (podem imaginar que és la regió de la finestra on pintarem els polígons).



● Punts vàlids

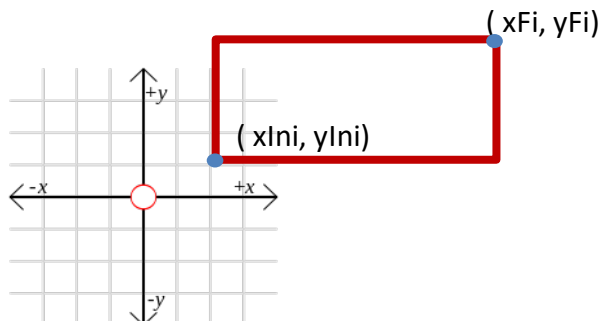
● Punts NO vàlids

La segona part de la pràctica 1 consistirà en crear una nova classe per a guardar un punt a més de la regió que els conté a tots. L'anomenarem **PuntRestringit** per diferenciar-la de la classe de la primera part.

Crearem aquesta classe en un nou projecte. Afegirem també una nova classe aplicació per validar el PuntRestringit.

El PuntRestringit tindrà els mateixos mètodes que el punt, adaptats a la nova lògica i les doses a tenir en compte són:

- Com guardem la regió rectangular? Per exemple amb els valors tal com es mostra al gràfic (x_{lni} , y_{lni}) i (x_{Fi} , y_{Fi})
- Tots els nous punts han de pertànyer a dins de la regió
- L'operació reflexió respecte l'eix X actuarà de la següent forma, farà la reflexió si el punt resultant queda dins de la regió, no la farà en cas contrari.
- La regió s'ha de poder ampliar (deixar igual o fer-la més gran, mai reduir-la per garantir que no quedarà cap punt fora d'ella).
- Hem de poder consultar en tot moment els límits d'aquesta regió des de les aplicacions que els utilitzin



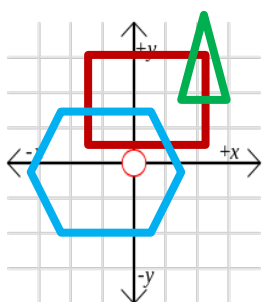
Valida la nova classe PuntRestringit amb una classe aplicació anomenada UsaPuntRestringit

Part 3. Afegir l'element Poligon com una llista de punts.

Ja disposem de la classe que permet representar l'element més simple, el punt. Tenim la solució en l'avaluació del taller de la primera part de la pràctica. En aquesta tercera part partirem d'aquesta solució, no tindrem en compte tot el treball fet en la part 2.

Ara volem poder representar qualsevol tipus de polígon. Un polígon és un conjunt de vèrtexs o punts en un ordre concret. Per tant, el nou element que hem de definir, la classe Poligon, ha de permetre guardar una llista d'objectes de la classe Punt.

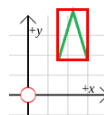
A la imatge exemple tenim un triangle, polígon de 3 punts, un rectangle amb 4 punts i un hexàgon amb 6 punts.



En aquesta tercera part es demana crear una nova classe Poligon que utilitzarà la classe Punt de la primera part i una nova classe UsaPoligon, aplicació que validarà les operacions de la classe Poligon.

La nova classe Poligon ha de tenir:

- Definició dels atributs per guardar la informació: array de punts amb mida fixada al constructor.
- Constructor que reserva espai per a guardar un número de vèrtexs determinat
- Setters/getters lògics
- Afegir un vèrtex al polígon
- Modificar el vèrtex i-èssim del polígon
- Consultar el vèrtex i-èssim del polígon
- toString
- Crear un duplicat del polígon
- Calcular i retornar el polígon reflexió sobre l'eix X – és un polígon on tots els seus vèrtexs s'han substituït per la seva reflexió sobre l'eix de les x
- Calcular i retornar el perímetre del polígon - El perímetre és la suma de les longituds de les arestes del polígon. La longitud d'una aresta es calcula com la distància entre els dos vèrtexs que la formen.
- Calcular la capsà contenidora del polígon - La capsà contenidora d'un polígon és el rectangle amb costats paral·lels als eixos de coordenades, de menor tamany que conté al polígon. En la següent figura es pinta la capsà contenidora del triangle.



La classe UsaPoligon és l'aplicació que validarà les operacions de la classe Poligon