

Exercícios/Atividade de Fixação

Discuta a interface com o professor, analise, projete e implemente os seguintes TADS:

- TAD data (um valor de data, com dia, mês e ano);
- TAD hora (um valor de horário, com hora, minuto e segundo);
- TAD numero complexo (um número complexo, com parte real e imaginária);
- TAD equação 2º grau, com raízes complexas;
- TAD ponto (um ponto no plano cartesiano, com coordenada x e y):
 - *criar(x,y)*
Retorna um tad ponto a partir das coordenadas x,y do ponto no plano cartesiano.
 - *distancia(paramTADptoA, paramTADptoB)*
Retorna um float com a distância euclidiana entre dois tad pontos.
 - *getXY(paramTADpto)*
Retorna uma tupla com as coordenadas x e y do ponto.
 - *igual(paramTADptoA, paramTADptoB)*
Retorna boolean True se os dois tad ponto de entrada forem iguais (mesmo valores de coordenadas), False caso contrário.
 - *move(paramTADpto, float dx, float dy)*
Desloca, no plano, o ponto de dx na coordenada x e de dy na coordenada y.
 - *intersec(paramTADretaA, paramTADretaB)*
Retorna um tad retângulo que é a intersecção dos dois retângulo de entrada, ou **None** se os retângulo de entrada não se intercederem.
- TAD reta (um reta no plano cartesiano):
 - *criar(x1,y1,x2,y2)*
Retorna um tad reta a partir das coordenadas x,y de dois pontos no plano cartesiano.
 - *criarP(paramTADptoA,paramTADptoB)*
Retorna um tad reta a partir de dois tad pontos de entrada.
 - *getPtos(paramTADreta)*
Retorna uma lista com os dois tad pontos usados na criação da reta.
 - *distancia(paramTADreta, paramTADpto)*
Retorna um float com a distância euclidiana entre os tad ponto e tad reta de entrada.
 - *coef(paramTADretaA)*
Retorna um float com o valor do coeficiente angular da reta (inclinação em relação ao eixo x).
 - *fy(paramTADretaA, float)*
Recebe uma reta e uma coordenada x de um suposto ponto pertencente à reta. Retorna o valor da coordenada y desse suposto ponto.
 - *pertence(paramTADretaA, paramTADpto)*
Retorna boolean True se o ponto de entrada pertence à reta de entrada, False caso contrário.
 - *igual(paramTADretaA, paramTADretaB)*

Retorna boolean True se as retas de entrada forem iguais, False caso contrário.

- *intersec(paramTADretaA, paramTADretaB)*
Retorna um tad ponto que é a intersecção dos dois tad reta. Retorna **None** se as retas de entrada não intercederem.
- *paralela(paramTADretaA, paramTADretaB)*
Retorna um boolean True se os dois tad reta de entrada são de retas paralelas entre si, False caso contrário.
- *perpendicular(paramTADretaA, paramTADretaB)*
Retorna um boolean True se os dois tad reta de entrada são de retas perpendiculares entre si, False caso contrário.

■ TAD circulo (um círculo no plano cartesiano):

- *criar(xcentro,ycentro,raio)*
Retorna um tad retângulo a partir da origem (canto superior esquerdo) e da extremidade direita inferior.
- *criarP(paramTADpto, raio): cria via origem e dimensões.*
Retorna um tad retângulo a partir da origem (canto superior esquerdo) e da largura e altura.
- *getCentro(paramTADcirc)*
Retorna um tad ponto com as coordenadas do centro do círculo.
- *getRaio(paramTADcirc)*
Retorna um float com o tamanho do raio usado para a criação do círculo.
- *perimetro(paramTADcirc)*
Retorna um float com o valor do perímetro.
- *area(paramTADcirc)*
Retorna um float com a área do retângulo.
- *igual(paramTADcircA, paramTADcircB)*
Retorna boolean True se os dois retângulos de entrada forem iguais (mesmo valores de lados), False caso contrário.
- *move(paramTADcircA, float dx, float dy)*
Desloca o círculo no plano (sua origem) de dx na coordenada x e dy na coordenada y.
- *contido(paramTADcircA, paramTADcircB)*
Retorna um boolean True se o círculo A está totalmente contido no círculo B, False caso contrário.

■ TAD retangulo (um retângulo no plano cartesiano):

- *criarVtx(xsupesq,ysupesq,xinfd,yinf): cria via vértices.*
Retorna um tad retângulo a partir da origem (canto superior esquerdo) e da extremidade direita inferior.
- *criarDim(xsupesq,ysupesq,larg,alt): cria via origem e dimensões.*
Retorna um tad retângulo a partir da origem (canto superior esquerdo) e da largura e altura.
- *getCantos(paramTADret):*
Retorna uma lista de tad pontos com os cantos superior esquerdo e inferior direito, respectivamente.
- *perimetro(paramTADret)*
Retorna um float com o valor do perímetro.
- *area(paramTADret)*
Retorna um float com a área do retângulo.
- *igual(paramTADretA, paramTADretB)*
Retorna boolean True se os dois retângulos de entrada forem iguais (mesmo valores de lados), False caso contrário.
- *move(paramTADretA, float dx, float dy)*

Desloca, no plano, o retângulo (sua origem) de dx na coordenada x e dy na coordenada y .

■ *intersec(paramTADretA, paramTADretB)*

Retorna um tad retângulo que é a intersecção dos dois retângulo de entrada, ou **None** se os retângulo de entrada não se intercederem.

■ TAD poligono (um polígono fechado no plano cartesiano);

■ TAD jogo-da-velha

■ TAD numero ip (um endereço ip de internet)

Obs: Construa uma aplicação simples e teste a interface e a funcionalidade de cada TAD.

Ou, construa um protótipo de aplicação primeiro e a partir daí analise qual seria a interface mais adequada para a construção do TAD.