

Interface Serializable	
<ul style="list-style-type: none"> Interface que os objetos serializáveis devem implementar para serem escritos em um arquivo std::string toJSON() void fromJSON() 	

Interface Object Serializable Point, Segment, Polygon	
<ul style="list-style-type: none"> Representa um objeto genérico (ponto, segmento ou polígono) void draw() Polygon getBoundingBox() double distance() 	<ul style="list-style-type: none"> Canvas

State Serializable	
<ul style="list-style-type: none"> Armazena o estado global de objetos e escalares definidos Responsável por manter o DAG (grafo direcionado acíclico) de dependências entre os objetos criados Processa remoções de objetos Recomputa os objetos necessários nas mudanças 	

Point Object	
<ul style="list-style-type: none"> Representa um ponto no plano double x double y Point operator + (const Point pt) Point operator - (const Point pt) Point operator * (const int c) long long operator * (const pt p) long long operator ^ (const pt p) 	

Segment		Object
<ul style="list-style-type: none"> • Representa um segmento de reta no plano • Point p • Point q • double length() • double angular_coefficient() • bool has(Point p) • bool intersects(Segment s) • int lattice_points() 		<ul style="list-style-type: none"> • Point

Polygon		Object
<ul style="list-style-type: none"> • Representa um polígono no plano • vector<Point> v • double area() • double perimter() • bool has(Point p) • int lattice_points() • Polygon operator + (const Polygon& p) // soma de minkowski 		<ul style="list-style-type: none"> • Point

Utils		
<ul style="list-style-type: none"> • long long sarea2(Point p, Point q, Point r) • long long collinear(Point p, Point q, Point r) • bool ccw(Point p, Point q, Point r) • int quad(Point p) • Polygon convex_hull(vector<Object> objects) • Polygon bounding_box(vector<Object> objects) 		<ul style="list-style-type: none"> • Point • Segment • Polygon • Canvas

Canvas	
<ul style="list-style-type: none"> • void drawBackground() • void drawAxis() • void drawObjects() • void draw() 	<ul style="list-style-type: none"> • Background • GeometryUtils • Point • Segment • Polygon

Menu	
<ul style="list-style-type: none"> • Criar e exibir graficamente um menu lateral • Listar objetos criado • Recebe e processa a entrada do usuário através de comandos • Exibir a definição de um objeto em texto 	<ul style="list-style-type: none"> • State • Canvas • Object

FileManager	
<ul style="list-style-type: none"> • Gerencia arquivos no sistema • Abre e escreve arquivos JSON contendo objetos do programa serializados • Abre e escreve imagens rasterizadas (png) e vetoriais (svg) 	<ul style="list-style-type: none"> • Canvas • Menu • State • Object

Interface	<div> <div>Background</div> <div>GridBackground GraphBackground PolarBackground</div> </div>
<ul style="list-style-type: none"> • Desenha o plano de fundo da aplicação • Ajusta o desenho do grid de acordo com o viewport e zoom atual 	<ul style="list-style-type: none"> • Canvas

GridBackground		Background
<ul style="list-style-type: none"> • Desenha um plano de fundo pontilhado 	<ul style="list-style-type: none"> • Canvas 	

GraphBackground		Background
• Desenha um plano de fundo quadriculado	• Canvas	

PolarBackground		Background
• Desenha um plano de fundo radial (cordenadas polares)	• Canvas	

