Class Diagram
Low Level

<<functions>>
Parser

parser(vector*): void
pointVectorReadIn(vector, int, ifstream): void
penInfo(ifstream,string,int,string,string, string): void
brushInfo(ifstream, string, string): void
setPenInfo(Shape*, string, int, string, string, string) : void
setBrushInfo(Shape*, string, string): void

<<class>>
Vector

INT_SIZE: int
size_v: int
elem: template
space: int

vector(): object
vector(int): object
vector(vector): object
operator=(vector): object
vector(vector&&): object
operator=(vector&&): object
~vector(): object
operator[](int): template
size(): int
capacity(): int
resize(int): void
push_back(template): void
reserve(int): void
begin(): const_iterator
begin(): *iterator
end(): const_iterator
end(): *iterator
insert(*iterator, template): *iterator
erase(*iterator): *iterator
copyFrom(vector): void
cleanUp(): void

<<class>>
Canvas

shapes: vector

Canvas(): object
~Canvas(): object
minimumSizeHint(): object
sizeHint(): object
paintEvent(QPaintEvent): void
addShape(): void
getNextId(): void

<<interface>>
Shape

pPainter: QPainter*
id: int
pen: Qpen
penColor: QColor
penStyle: PenStyle
penCapStyle: PenCapStyle
penJoinStyle: PenJoinStyle
brush: QBrush
brushColor: QColor
brushStyle: BrushStyle
dims: vector<int>
penWidth: int

0..n

<<class>>
Rectangle

isSquare: bool

1 Rectangle(int, bool): object

<<class>> Ellipse

---

isCircle: bool

Ellipse(int, bool): object
paintEvent(QPaintEvent*): void

<<class>> Line

Line(int): object
paintEvent(QPaintEvent*): void

Shape(int): object
~Shape(): object
updateDimensions(vector<int>): void
updateDimensions(string): void
paintEvent(QPaintEvent*): void
setPenColor(QColor): void
setPenColor(string): void
setPenWidth(int): void
setPenWidth(string): void
setPenStyle(PenStyle): void
setPenStyle(string): void
setPenCapStyle(PenCapStyle): void
setPenCalStyle(string): void
setPenJoinStyle(PenJoinStyle): void
setPenJoinStyle(string): void
setBrushColor(QColor): void
setBrushColor(string): void
setBrushStyle(BrushStyle): void
setBrushStyle(string): void
makeShape(string, string): object*
makeShape(int, Shapes): object*
getId(): int
getPen(): QPen&
getBrush(): QBrush&
drawId(QRect): void

---

paintEvent(QPaintEvent*): void

<<class>> Text

---

textString: string
alignment: AlignmentFlag
pointSize: int
fontFamily: string
fontStyle: Style
fontWeight: Weight

Text(int, bool): object
paintEvent(QPaintEvent*): void
setTextString(string): void
setAlignment(AlignmentFlag): void
setAlignment(string): void
setPointSize(int): void
setPointSize(string): void
setFontFamily(string): void
setFontStyle(Style): void
setFontStyle(string): void
setFontWeight(Weight): void
setFontWeight(string): void
getFont(): object

<<class>> Polygon

Polygon(int): object
paintEvent(QPaintEvent*): void

<<class>> Polyline

PolyLine(int): object
paintEvent(QPaintEvent*): void