



<<class>>  
**QDialog**  
Qt Dialog Object

<<class>>  
**logindialog**

-\*ui: logindialog  
-isLoggedIn: boolean

+<<constructor>> logindialog(\*parent:QWidget=nullptr): explicit  
+<<destructor>> ~logindialog()  
+getIsLoggedIn(): boolean  
-on\_pushButton\_clicked(): void

<<class>>  
**ContactUs**

-\*ui: ContactUs

+<<constructor>> ContactUs(\*parent:QWidget=nullptr): explicit  
+<<destructor>> ~ContactUs()

<<class>>  
**comments**

-\*ui: commens  
-isCustomer: bool

+<<constructor>> comments(\*parent:QWidget=nullptr): explicit  
+<<destructor>> ~comments()  
-on\_pushButton\_clicked(): void  
-on\_pushButton\_2\_clicked(): void

<<class>>  
**ShapeListing**

-\*ui: ShapeListing  
-areaVec: Shape\*[\*]  
-periVec: Shape\*[\*]

+<<constructor>> ShapeListing(\*parent:QWidget, shapeVec:Shape\*[\*]): explicit  
+getShapeName(shape:ShapeType): QString const  
+<<destructor>> ~ShapeListing()  
compPerimeter(\*i:Shape,\*j:Shape): bool  
compArea(\*i:Shape,\*j:Shape): bool

<<class>>  
**addShape**

-\*ui: addShape  
+\*newShape: Shape  
+addingShapeID: int

+<<constructor>> addShape(\*parent:QWidget=nullptr, &shapeCountFromMain:const int=0): explicit  
  
+getShapeCount(): int const  
+addShapeToCanvas(): void  
+addLine(): void  
+addPolyline(): void  
+addPolygon(): void  
+addRectangle(): void  
+addSquare(): void  
+addEllipse(): void  
+addCircle(): void  
+addText(): void  
+\*getNewShape(): Shape const  
+<<Destructor>> ~addShape()  
-on\_button\_accepted(): void  
-getStringColor(): GlobalColor  
-getStringFlag(): AlignmentFlag  
-getTextFontFamily(): QString  
-getBrushColor(): GlobalColor  
-getPenCapStyle(): PenCapStyle  
-getPenStyle(): getPen  
-getPenJointStyle(): PenJoinStyle  
-getBrushStyle(): BrushStyle  
-getFontStyle(): Style  
-getFontWeight(): Weight

<<class>>  
**deleteshape**

-\*ui: deleteshape  
-toDelete: int  
-shapeCount: int

+<<constructor>> deleteshape(\*parent:QWidget=nullptr, &shapeCountFromMain:const int=0, &shapeVec:Shape\*[\*]): explicit  
  
+<<destructor>> ~deleteshape()  
+getShapeCount(): int  
+getToDelete(): int  
+getShapeName(shape:ShapeType): QString  
-on\_buttonBox\_accepted(): void

<<class>>  
**ModifyShapes**

-\*modShape: Shape  
-indexModShape: int  
-\*ui: ModifyShapes  
-localVec: Shape\*[\*]

+<<constructor>> ModifyShapes(\*parent:QWidget, ShapeVec:Shape\*[\*])  
+<<destructor>> ~ModifyShapes()  
+\*getModShape(): Shape const  
+getModIndex(): int const  
-on\_shapesComboBox\_currentIndexChanged(index:int): void  
-on\_buttonBox\_accepted(): void  
-getShapeName(shape:ShapeType): QString const  
-disableAll(): void const  
-enableAll(shape:ShapeType): void const  
-addShapeToCanvas(): void  
-addLine(): void  
-addPolyline(): void  
-addPolygon(): void  
-addRectangle(): void  
-addSquare(): void  
-addEllipse(): void  
-addCircle(): void  
-addText(): void  
-getStringColor(): GlobalColor  
-getStringFlag(): AlignmentFlag  
-getTextFontFamily(): QString  
-getBrushColor(): GlobalColor  
-getShapeType(shape:QString): ShapeType  
-getColor(): GlobalColor  
-getFontColor(): GlobalColor  
-getPenCapStyle(): PenCapStyle  
-getPenStyle(): PenStyle  
-getPenJointStyle(): PenJoinStyle  
-getBrushStyle(): BrushStyle  
-getFontStyle(): Style  
-getFontWeight(): Weight

<<class>>  
**RenderArea**  
Qt Class

-renderArea: const QImage  
-ShapeMagazine: Shape[\*]  
-numShapes: int

+<<constructor>> RenderArea(parent:QWidget)  
+paintEvent(event:QPaintEvent): void  
+<<override>> sizeHint(): QSize const  
+minimumSizeHint(): QSize const  
+getShapes(): Vector<Shape> const  
+addShape(shapeIn:Shape): void  
+getSize(): int  
+getNumShapes(): int  
+chopShape(indexRemove:int): void  
+moveShape(indexMove:int,coordMove:int,x:int,y:int): void  
-readShapeFile(): void  
-getStringColor(color:QColor): QString  
-getStringPenCap(penCapStyle:PenCapStyle): QString  
-getStringPenStyle(penStyle:PenStyle): QString  
-getStringPenJointStyle(penJoinStyle:PenJoinStyle): QString  
-getStringBrush(brush:BrushStyle): QString  
-getStringFlag(flag:AlignmentFlag): QString  
-getStringFontStyle(fontStyle:QFont): QString  
-getStringFontWeight(fontWeight:int): QString  
-getShapeType(QString shape): Shapes  
-getColor(color:QString): GlobalColor  
-getPenCapStyle(cap:QString): PenCapStyle  
-getPenStyle(pen:QString): PenStyle  
-getPenJoinStyle(penJoint:QString): PenJoinStyle  
-getBrushStyle(brushStyle:QString): BrushStyle  
-getFlag(flag:QString): AlignmentFlag  
-getFontStyle(fontStyle:QString): Style  
-getFontWeight(fontWeight:QString): Weight