

```

-----
# OnPreparePrinting(CPrintInfo* pInfo): BOOL
# OnBeginPrinting(CDC* pDc, CPrintInfo* pInfo): void
# OnEndPrinting(CDC* pDc, CPrintInfo* pInfo): void
# OnPrepareDC(CDC* pDC, CPrintInfo* pInfo = (CPrintInfo*)0): void
# OnPrint(CDC* pDC, CPrintInfo* pInfo): void
-----

# OnCreate(LPCREATESTRUCT lpcs): int
# OnInitialUpdate(): void
# OnSize(UINT nType, int cx, int cy): void
-----

# OnHScroll(UINT nSBCode, UINT nPos, CScrollBar* pScrollBar): void
# OnLButtonUp(UINT nFlags, CPoint point): void
# OnLButtonDown(UINT nFlags, CPoint point): void
# OnLButtonDblClk(UINT nFlags, CPoint point): void
# OnMouseMove(UINT nFlags, CPoint point): void
# OnMouseWheel(UINT nFlags, short zDelta, CPoint pt): BOOL
# OnRButtonUp(UINT nFlags, CPoint point): void
# OnRButtonDown(UINT nFlags, CPoint point): void
# OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags): void
# OnKeyUp(UINT nChar, UINT nRepCnt, UINT nFlags): void
-----

# OnContextMenu(CWnd* pWnd, CPoint point): void
# OnCopy(): void
# OnPaste(): void
-----

# DECLARE_MESSAGE_MAP()
-----

- expandLineBounds(): void
- selectClippingRgn(int nAction): void
- drawHeader(CDC* pDc): void

```

MFC
CPropertySheet



CSmartColour

types: CSmartColourTypes

statements: CSmartColourStatements

builtin: CSmartColourBuiltin

other: CSmartColourOther

+ CSmartColour(std::vector<std::vector<CString>> page_strings, std::vector<COLORREF> page_colour, CString commentType, COLORREF commentColour, COLORREF numberColour, COLORREF highlightColour, CString fonts, CString indents, LPCTSTR pszCaption = L"", CWnd* pParentWnd = nullptr, UINT iSelectPage = 0)

+ ~CSmartColour()

DECLARE_DYNAMIC(CSmartColour)

DECLARE_MESSAGE_MAP()

MFC
CMFCPropertyPage



CSmartColourTypes

----- #ifdef AFX_DESIGN_TIME

+ enum { IDD = IDD_SMARTCOLOUR_TYPES}

+ m_Listbox_Types_control: CVSListBox

+ m_Combo_Types_colour_control: CMFCColorButton

+ m_Combo_Types_colour_var: COLORREF

+ m_listbox_types_strings: std::vector<CString>

+ CSmartColourTypes(std::vector<CString> types_string, COLORREF colour, CWnd* pParent = nullptr)

+ ~CSmartColourTypes()

DoDataExchange(CDataExchange* pDX): void

OnInitDialog(): BOOL

OnKillActive(): BOOL

DECLARE_DYNAMIC(CSmartColourTypes)

DECLARE_MESSAGE_MAP()

CSmartColourStatements

----- #ifdef AFX_DESIGN_TIME

+ enum { IDD = IDD_SMARTCOLOUR_OPS}

+ m_Listbox_Statements_syntax_control: CVSListBox

+ m_Combo_Statements_syntax_control: CMFCColorButton

+ m_Listbox_Statements_statments_control: CVSListBox

+ m_Combo_Statements_statments_control: CMFCColorButton

+ m_Strings_Statements_syntax_var: std::vector<CString>

+ m_Colour_Statements_syntax_var: COLORREF

+ m_Strings_Statements_statements_var: std::vector<CString>

+ m_Colour_Statements_statements_var: COLORREF

+ CSmartColourStatements(std::vector<CString> syntax_string, COLORREF syntax_colour, std::vector<CString> statement_string, COLORREF statement_colour, CWnd* pParent = nullptr)

+ ~CSmartColourStatements()

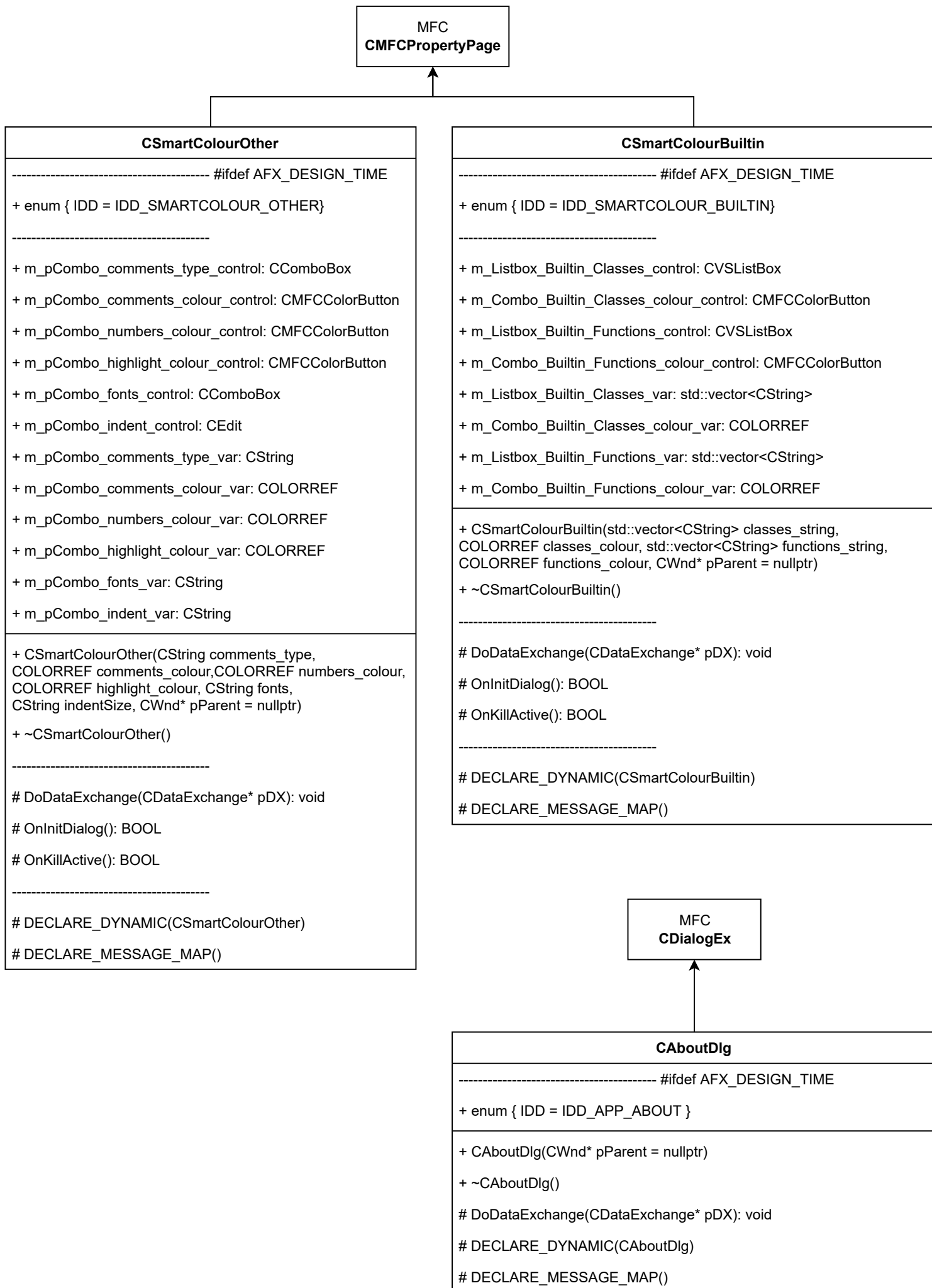
DoDataExchange(CDataExchange* pDX): void

OnInitDialog(): BOOL

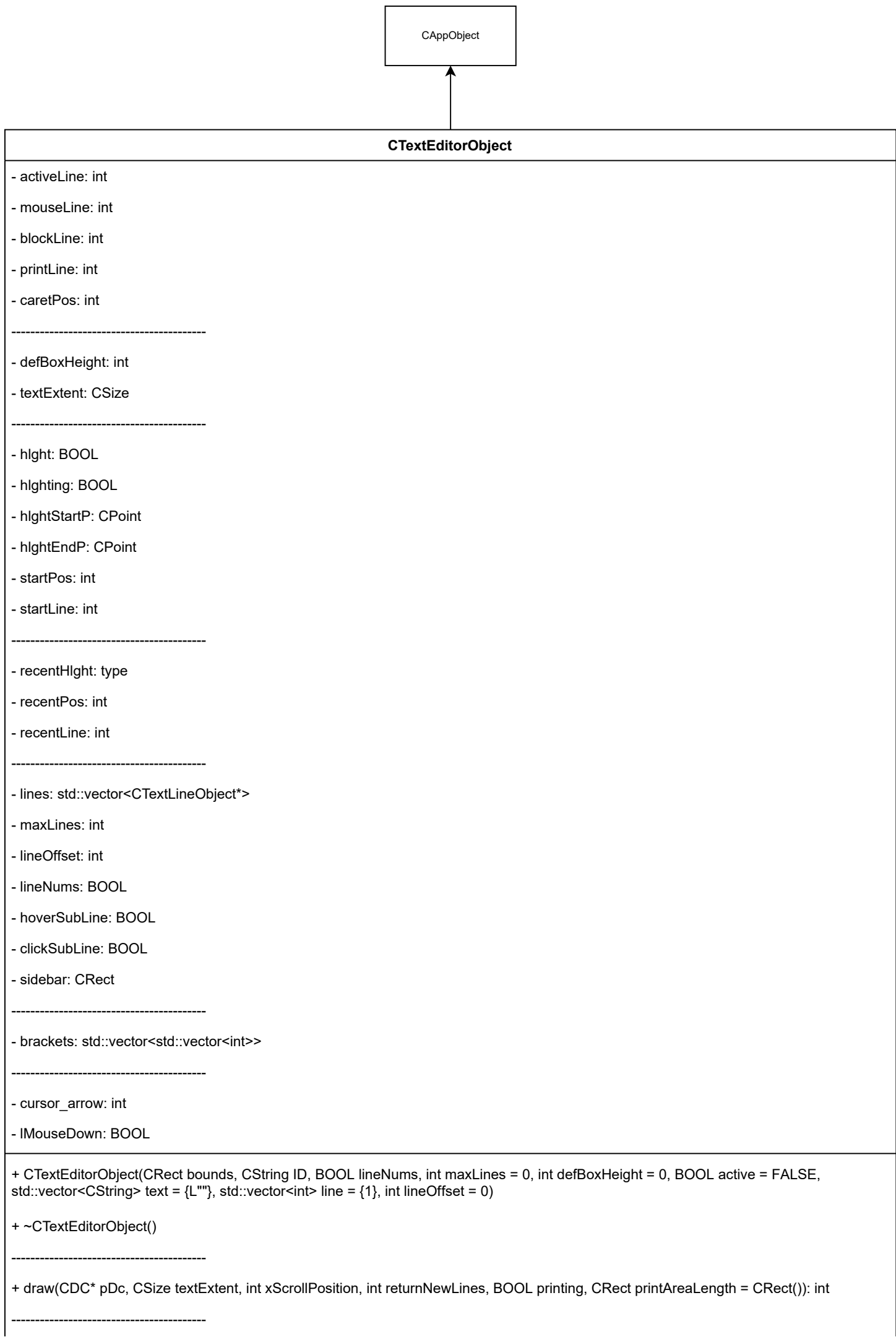
OnKillActive(): BOOL

DECLARE_DYNAMIC(CSmartColourStatements)

DECLARE_MESSAGE_MAP()



CAppObject
ID: CString # bounds: CRect # active: BOOL
+ CAppObject(CRect bounds, CString ID, BOOL active = FALSE) + ~CAppObject() ----- + draw(CDC* pDc): void ----- + OnSize(UINT nType, int cx, int cy): void + OnLButtonUp(UINT nFlags, CPoint point): void + OnLButtonDown(UINT nFlags, CPoint point): void + OnLButtonDblClk(UINT nFlags, CPoint point): BOOL + OnRButtonUp(UINT nFlags, CPoint point): void + OnRButtonDown(UINT nFlags, CPoint point): void + OnMouseMove(UINT nFlags, CPoint point): BOOL ----- + OnRecieveText(CString text): int + OnRecieveBackspace(): void + OnRecieveReturn(): void + OnRecieveTab(): void ----- + OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags): void + OnContextMenu(CWnd* pWnd, CPoint point): BOOL ----- + getID(): CString + setID(CString ID): void + getBounds(): CRect + checkPointInBounds(CPoint point): BOOL + setBounds(CRect bounds): void + setBounds(int x, int y, int cx, int cy): void + setPosition(CPoint point): type + move(int x, int y): void + isActive(): BOOL + setActive(): void



```

+ getCaretPos(): int
+ getCaretPoint(CSize caretSize): CPoint
+ getTextExtent(): CSize
+ setTextExtent(CSize size): void
+ getRecentPos(): int
+ pointHighlighted(CPoint point): BOOL
+ hasHighlight(): BOOL
+ hlghtingOff(): void
+ getHighlightClippingRect(): CRect
+ getHighlightExactRgn(int x_offset, int y_offset): CRgn*
+ getStartLine(): int
+ isHlghtMultiline(): BOOL
+ lineHighlight(int line): int
+ iGetLineNum(int a = 0): std::vector <int>
+ sGetLineNum(int a = 0): CString
+ incrementSublines(int subline, int val): void
+ getActiveLine(): int
+ getBlockLine(): int
+ getPrintLine(): int
+ getLineText(int line = 0): CString
+ getLineBounds(int line = 0): CRect
+ getHighlightedText(): CString
+ getLineTextWidth(): int
+ getNumLines(): int
+ getBoxHeight(BOOL default = FALSE): int
+ setActive(BOOL active): void
+ move(int x, int y): void
+ setBounds(CRect bounds): void
+ getPrintBounds(int returnNewLines, int printAreaLength): std::tuple<CRect, int>
+ getCursorArrow(): int
-----
+ initialise(): void
-----

+ OnSize(UINT nType, int cx, int cy): void
+ OnLButtonUp(UINT nFlags, CPoint point): void
+ OnLButtonDown(UINT nFlags, CPoint point): void
+ OnLButtonDbIClk(UINT nFlags, CPoint point): BOOL
+ OnRButtonUp(UINT nFlags, CPoint point): void
+ OnRButtonDown(UINT nFlags, CPoint point): void
+ OnMouseMove(UINT nFlags, CPoint point): BOOL
-----
+ OnRecieveText(CString text): int
+ OnRecieveBackspace(): void
+ OnRecieveReturn(): void
+ OnRecieveTab(): void

```

+ OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags): void

+ OnContextMenu(CWnd* pWnd, CPoint point): BOOL

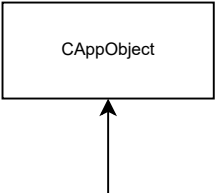
- setSidebar(): void

- setHighlighter(): void

- setBracketsNull(): void

- bracketContains(int value, int& itPos, int type, int side = 1): BOOL

- moveBrackets(int val, int index): void



CTextLineObject
<div>- text: CString</div> <div>- highlighter: Gdiplus::Rect</div> <div>- highlight: BOOL</div> <div>- hStart: int</div> <div>- hEnd: int</div> <div>- lineHighlight: BOOL</div> <div>- numSubLines: int</div> <div>- lineNums: std::vector<int></div> <div>- smartColour: BOOL</div>
<div>+ CTextLineObject(CRect bounds, CString ID, std::vector<int> lineNums, CString text = L"", BOOL active = FALSE)</div> <div>+ ~CTextLineObject()</div> <div>-----</div> <div>+ draw(CDC* pDc, CSize textExtent, std::vector<std::vector<int>> brackets, int returnNewLines, BOOL printing, int printAreaLength = -1): int</div> <div>-----</div> <div>+ getText(): CString</div> <div>+ getLength(): int</div> <div>+ setText(CString text): void</div> <div>+ concatenateString(CString text, int position): void</div> <div>+ backspace(int position): void</div> <div>+ isHighlighting(): BOOL</div> <div>+ setHighlighting(BOOL val): void</div> <div>+ setHighlighter(int pos1, int pos2): void</div> <div>+ highlightLine(): void</div> <div>+ isLineHighlighted(): BOOL</div> <div>+ getHStart(): int</div> <div>+ getHEnd(): int</div> <div>+ getHighlightedText(): CString</div> <div>+ getNumSubLines(): int</div> <div>+ iGetLineNums(): std::vector<int></div> <div>+ sGetLineNums(): CString</div> <div>+ addSublines(std::vector<int> subs): void</div> <div>+ incrementLine(int subline, int val): void</div> <div>+ setBounds(CRect bounds): void</div> <div>+ move(int x, int y): void</div> <div>-----</div> <div>- onlyNums(CString str): BOOL</div> <div>- brkPosContains(std::vector<std::vector<int>> vector, int value, int side = 1): BOOL</div>

```
1
2 // To meet the SQA Advanced Higher Computing Science criteria,
3 // the program implements a insertion sort in all
4 // CSmartColour's property pages when closing them
5
6 // The following pseudocode is a design for CSmartColourTypes' sort
7 // It will show how it works in practise for the rest of the property
  sheets
8
9 FOR i IN RANGE(1, m_Listbox_Types_control.SizeOf()):
10
11     CREATE variables
12
13     WHILE pos > 0 AND m_Listbox_Types_control[pos-1] > currentValue:
14         SET m_Listbox_Types_control[pos] TO m_Listbox_Types_control[po
s - 1]
15         pos -= 1
16
17     m_Listbox_Types_control = currentValue
18
19
11.1 DECLARE CString currentValue INITIALLY AS m_Listbox_Types_control[i]
11.2 DECLARE INT pos INITIALLY AS i
```