

CLASSPAD MENGEN ADD-IN

VERSION 1.0



Bearbeitet 2011 für die HTW Dresden von:

Martin Pritzke

Christian Kettmann

Andreas Kahlenbach

INHALT

Verwandte Projekte	3
Lösungsansatz	3
Funktionalität	3
Bedienoberfläche	3
Alternative Bedienoberfläche aus früheren Versionen	4
Geplante Erweiterungen	5
Add-in Programmierung.....	6
Informationen für Add-In Programmierer	6
Portable Embedded GUI.....	6
PegThing.....	6
PegPresentationManager	6
PegMessageQueue.....	7
PegScreen.....	7
Mengen Add-In.....	7
CList.h / CList.cpp	7
CPKeyboardManager.h	7
globalvar.h.....	7
MainWindow.h / MainWindow.cpp.....	7
MengeKeypad.h / MengeKeypad.cpp.....	8
mengeninf.h / mengeninf.cpp.....	8
Weitere zentrale Funktionen für die Berechnungen	8
PEGMMAIN.HPP / Pegmain.cpp	8
Messages und Signale	9
Probleme während der Entwicklung.....	11
Anhang	12
A: Klassenliste (unvollständig).....	12
B: Classpad Spezifische Klassen.....	16
C: Sonstige Klassen	19
Quellen	21

Im Rahmen eines Projektseminars, welches von Prof. Paditz geleitet wurde, sollte unter anderem eine Add-In-Anwendung entwickelt werden. Mit dieser sollte es möglich sein, grundlegende Rechnungen mit Mengen durchführen zu können. Bei der Recherche über Software mit ähnlichen Zielstellungen fiel uns schnell auf, dass keine dieser Programme mit infiniten Mengen umgehen konnte. Dies motivierte uns ein Add-In ohne diese Einschränkung zu schreiben.

VERWANDTE PROJEKTE

Eine themenbezogene Suche über einen großen Online-Suchdienst ergab sehr viel Treffer aber wenig Verwertbares. Eine Ausnahme stellt der mächtige Onlinerechner Tusanga¹ dar. Dieser beherrscht neben einer erweiterten Funktionalität eines Taschenrechners die elementaren Rechenoperationen für finite Mengen. Eine weitere nennenswerte Software ist ein Zusatzpaket für den HP 50g Taschenrechner, welche von „Heuson-Software“² aus Deutschland entwickelt und kostenpflichtig vertrieben wird.

Das Classpad Add-In soll auch mit Mengen in infiniter Form umgehen können. Dafür wurde über die Zeichenkettenverarbeitung hinaus ein Konzept entworfen, welches arbeitsspeicherschonend und performant ist.

LÖSUNGSANSATZ

Am Ende des Grobentwurfes stand fest, dass das Problem am besten über Listen von Bereichen zu lösen ist. Eine Liste repräsentiert dabei eine Menge. Ein Bereich enthält sowohl die Grenzen, als auch deren Zugehörigkeit (eingeschlossen, ausgeschlossen, unendlich). Bei der Berechnung werden alle Bereiche aus allen Listen nacheinander, wenn durch Klammerung nichts anderes vorgegeben, von links nach rechts vereinfacht und laut Operation verrechnet. Als Resultat entsteht eine Ergebnisliste, die bestmöglich vereinfacht und der Größe nach von klein nach groß sortiert ist.

FUNKTIONALITÄT

Dem Nutzer stehen folgende Mengenoperationen zur Verfügung:

- Schnittmenge
- Vereinigungsmenge
- Differenzmenge
- Komplementärmenge

Folgende Mengen sind über ein Symbol vordefiniert:

- Leere Menge: \emptyset
- Allumfassende Menge: Ω

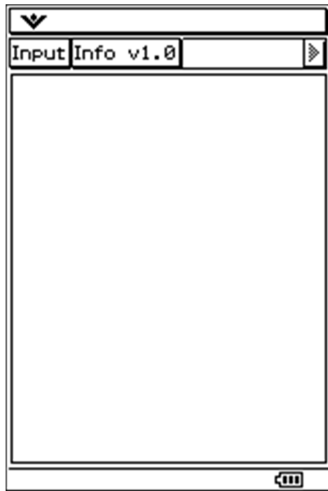
BEDIENBEREICH

Als Schnittstelle zum Nutzer steht ein einfaches Interface zur Verfügung. Über die Schaltfläche „Input“ in der Symbolleiste kann man die Eingabemengen definieren. Dafür steht eine extra auf die Bedürfnisse der Anwender zugeschnittene Tastaturebene zur Verfügung, mit deren Hilfe man schnell und bequem auf alle für das Mengen-Add-In relevanten Eingabezeichen zurückgreifen kann. Nach der Bestätigung des eingegebenen Ausdruckes über „OK“ wird dieser mit dem Ergebnis angezeigt. Über das erneute Drücken der Schaltfläche „Input“ kann der letzte Ausdruck editiert und eine erneute Berechnung instruiert werden.

¹ Adam Williams: Tusanga calculator (Februar 2009) <http://www.tusanga.com/cgi-bin/calc> (Stand: 26.02.2011).

² Clemens Heuson: HP50G HEUSON-SOFTWARE (November 2010) <http://heuson-software.de/heusonger.htm> (Stand: 26.02.2011).

Nach dem Start



Angepasste Tastatur



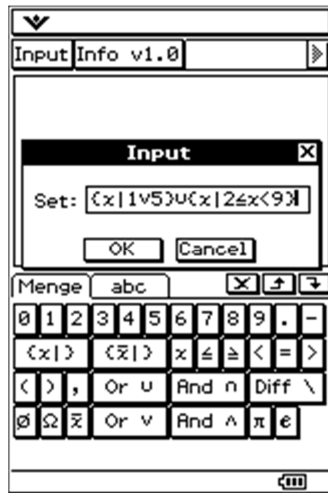
Infofenster



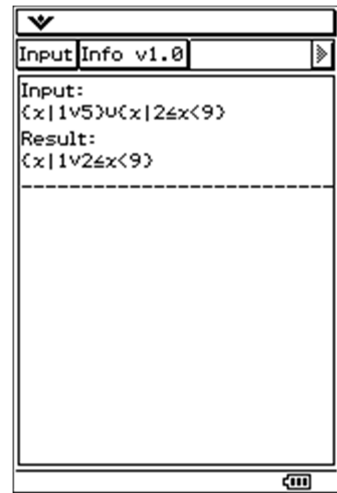
Ausdruckseditor vor der Eingabe



Ausdruckseditor nach der Eingabe



Ergebnisfenster



ALTERNATIVE BEDIENOBERFLÄCHE AUS FRÜHEREN VERSIONEN

Programmfenster



Diese Alternative Oberfläche besteht aus einer Editbox die zu Eingabe und Ausgabe verwendet wird. Die Eingabe muss auf einer Zeile geschrieben werden, Die Ausgabe erfolgt 2 Leerzeichen nach innen versetzt. Ist die Eingabe Fehlerhaft, so erscheint die Ausschrift "fehlerhafte Eingabe! Position: ??". Ein großer Nachteil ist, dass die Eingabezeile und Ausgabezeile immer abwechselnd untereinander folgen müssen. Hält man diese Reihenfolge nicht ein, so gibt der Interpreter einen Fehler zurück da er immer die übernächste Zeile, von der letzten eingelesenen Zeile aus gesehen, einliest (beginnend mit der Ersten). Ein Interner Zähler zählt wie oft der "berechnen"-Button geklickt wurde und erhöht sich jedes Mal um zwei. (1, 3, 5, ...)

Eine geeignete Oberfläche wäre eine, die wie das Main-Menu aufgebaut ist. In dieser werden Eingabe und Ausgabe in separaten "ScrollableMathWindow" angezeigt. Ein fertiges Objekt, welches wie das Main-Menu funktioniert und aufgebaut ist, ist nicht dokumentiert. So müsste es selber erstellt werden. Dazu ist eine Liste der von ScrollableMathWindow-Objekten zu erstellen. Außerdem müsste eine Methode gefunden werden, wie man im Quelltext das ScrollableMathWindow ermitteln kann, in welchem gerade der Cursor steht. Hat man das hinbekommen, so kann man die Berechnung auch durch das Drücken der EXE-Taste auslösen.

GEPLANTE ERWEITERUNGEN

Die nachfolgenden Ideen konnten auf Grund fehlender Zeit nicht mehr realisiert werden:

- Möglichkeit mit Variablen zu arbeiten:
Es wäre denkbar einzelne Mengen oder auch ganze Ausdrücke durch Variablen zu repräsentieren.
- Templates für Mengenergebnisse in Verbindung mit Variablen:
Beispiele:
 - Symmetrische Differenz zweier Mengen (A,B): $(A \cup B) \setminus (A \cap B)$
 - Prüfen ob eine Menge (A) Teilmenge von einer Menge B ist: $(A \cup B) = B$
 - Vergleichen zweier Mengen (A,B) auf Gleichheit: $(A \setminus B) \cup (B \setminus A) = \emptyset$
 - Prüfen ob ein Wert (W) in einer Menge (B) definiert ist: $(\{W\} \cap B) \neq \emptyset$
- Freigeben des nicht mehr benötigten Speicherplatzes (z.B.: Listen abbauen)

ADD-IN PROGRAMMIERUNG

Es gibt grundsätzlich zwei Möglichkeiten das Classpad mit benutzerdefinierten Funktionalitäten zu erweitern. Die wohl schnellste ist die Build-In Basic-Skriptsprache. Mit dieser lassen sich schnell viele Probleme lösen. Ist ein Problem besonders zeit- und/oder ressourcen-kritisch oder übersteigt eine gewisse Komplexität, sollte man eher ein Add-In entwickeln. Der Vorteil besteht darin, dass das Programm nicht zur Laufzeit interpretiert werden muss, sondern zuvor von einem Compiler-Maschinencode übersetzt wird. Eine weitere Möglichkeit den Classpad zu steuern, ist die Skriptsprache LUA in Verbindung mit einem Add-In. Das Add-In stellt einen LUA-Compiler bereit, der das LUA-Programm zunächst in einen Zwischencode übersetzt (kein Maschinencode) und danach in einer virtuellen Maschine ausführt. Von der Performance sind solche Programme zwar schneller als Basic-Skripte aber langsamer als eine reine Add-In Anwendung.

Einige Internetseiten, wie zum Beispiel www.casiocalc.org oder casigo.de, sammeln Add-In Anwendungen und stellen diese, teilweise mit Quellcode, zum Download bereit.

INFORMATIONEN FÜR ADD-IN PROGRAMMIERER

Der Classpad besitzt eine CPU von Hitachi mit der Bezeichnung „SH7291“. Dieser RISC-Prozessor hat 32 MHz und eine Befehlsbreite von 32 Bit. Alle Programme teilen sich 512 KB RAM. Für die Firmware und persistent gespeicherten Daten steht dem Classpad ein 5,3MB großer Flash-ROM zur Verfügung.

Es sollten keine Double- oder Float-Datentypen verwendet werden. Das Classpad bietet speziell dafür den Datentyp OBCD und für komplexe Zahlen CBCD.

PORTABLE EMBEDDED GUI

Die Benutzerschnittstelle basiert auf der universellen Library PEG (Portable Embedded GUI). Welche CASIO speziell auf den Classpad zugeschnitten hat, indem sie um spezifische Klassen erweitert wurde (siehe Anhang Abschnitt B und C). Dabei hat CASIO auf die Module wie „Image Conversions“, „Charting Classes“ und „HMI Classes“ verzichtet.

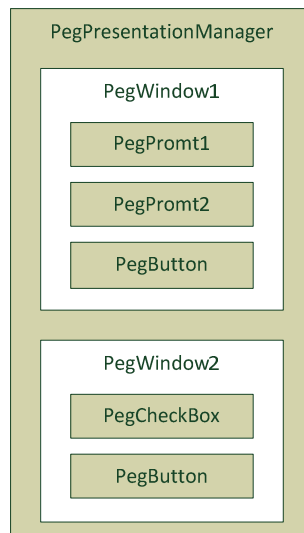
PEGTHING

PegThing ist die Klasse, von der alle PEG-Objekte abgeleitet sind. Sie stellt den kleinsten gemeinsamen Nenner zu allen PEG-Objekten dar und beinhaltet viele fundamentale Funktionen und Informationen. Zum Beispiel die Größe, den Style eines Objektes oder die Funktion Message(). Außerdem beinhaltet ein aus einem PegThing abgeleitetes Objekt drei wichtige Funktionen: Presentation(), MessageQueue(), Screen().

Eine umfangreiche Liste aus Standardobjekten, welche von PegThing abgeleitet sind, ist im Anhang zu finden.

PEGPresentationMANAGER

Der PegPresentationManager hat die Aufgabe, alle Fenster und deren Objekte zu koordinieren, um Tastatureingaben oder andere Interaktionen mit den Objekten zu ermöglichen.



PEGMESSAGEQUEUE

Die PegMessageQueue hält Messages bis zu deren Abarbeitung bereit. Eine mögliche Abarbeitungsroutine ist die Funktion Message().

PEGSCREEN

PegScreen ist eine Klasse, die jene Routinen beinhaltet, die ein Objekt braucht, um sich selber zu zeichnen. Sie isoliert alle Funktionen die in den Videospeicher schreiben. Um PEG an ein neues Gerät anzupassen, muss nur diese Klasse individuell angepasst werden.

MENGEN ADD-IN

CLIST.H / CLIST.CPP

Diese Klasse stellt die Funktionalität einer doppeltverketteten Liste zur Verfügung.

In diesem Projekt wird sie verwendet um zum Beispiel die „Bereiche“ zu verwalten.

CPKEYBOARDMANAGER.H

Enthält grundlegende Funktionen rund um das „virtuelle“ Keyboard.

GLOBALVAR.H

Diese Headerdatei wird in die „mengeninf.h“ eingebunden und gehört somit zum Programm-Core. Sie enthält einige globale Puffer.

MAINWINDOW.H / MAINWINDOW.CPP

Diese Klasse beschreibt das äußere Erscheinungsbild des Add-Ins und reagiert auf Ereignisse in der Symbolleiste.

MENGEKEYPAD.H / MENGEKEYPAD.CPP

Dieses Modul erzeugt das angepasste „virtuelle“ Keyboard und sorgt für die Ereignisbehandlung bei Buttonklick.

MENGENINF.H / MENGENINF.CPP

Dieses Modul beinhaltet die eigentliche Programmlogik (Core).

Die Funktion *bool mengen(CList* list, char end)*, direkt aufgerufen aus der Funktion *bool Parser()* ist der Einstiegspunkt für die Mengenberechnung. Sie liest die Eingabe aus dem Eingabepuffer und koordiniert weitere Funktionsaufrufe, darunter auch rekursive Aufrufe von sich selbst. Am Ende wird über den Parameter „list“, die Ergebnisliste zurückgegeben. Diese enthält nur noch Bereiche, die ODER verknüpft sind.

WEITERE ZENTRALE FUNKTIONEN FÜR DIE BERECHNUNGEN

tcompare **bereichCompare**(tbereich* bereich1, tbereich* bereich2):

Ermittelt in Bezug auf „bereich1“ eine Überlappung (contain) mit „bereich2“ oder ob dieser komplett außerhalb von „bereich1“ liegt. (less, greater)



void **bereichListOr**(CList* list1, CList* list2):

„list2“ wird an „list1“ angehängt, ohne diese danach zu vereinfachen.

CList* **bereichListAnd**(CList* bereichList1, CList* bereichList2):

Jede Liste mit Bereichen (Mengen) wird zunächst maximal über die Funktion **bereichListSimplify()** vereinfacht. Danach können die Mengen mit Hilfe der Funktion **bereichAnd()** verknüpft werden. Als Ergebnis dieser Funktion entsteht eine Ergebnisliste.

CList* **bereichListMinus**(CList* bereichList, CList* bereichListMinus):

Bei der Differenzmenge werden die Eingangsmengen zunächst vereinfacht. Von „bereichList“ werden überlappende Bereiche mit „bereichListMinus“ abgezogen. Dies erledigt die Funktion **bereichMinus()** welche wiederum **bereichAnd()** nutzt um den Bereich zu ermitteln, welcher von dem ursprünglichen Bereich abgezogen werden muss. Das Ergebnis ist die Differenzmenge.

void **listToString**(CList* list):

Diese Funktion erzeugt aus einer Liste von Bereichen (eine Menge) eine Textausgabe.

PEGMAIN.HPP / PEGMAIN.CPP

In diesem Modul wird das Add-In initialisiert. Das Hauptfenster wird erzeugt und das Keyboard eingerichtet.

MESSAGES UND SIGNALE

Mit Hilfe von Messages ist es leicht möglich auf Veränderungen im System zu reagieren. Zum Beispiel, wenn der Benutzer einen Button drückt oder ein Timer abgelaufen ist.

Um eine Message abfangen zu können, muss die Methode `Message()` aus `PegThing` überschrieben werden:

```
SIGNED Message(const PegMessage &Mesg);
```

Die Struktur `PegMessage` beinhaltet alle relevanten Informationen, die im Zusammenhang mit einem Ereignis stehen. Die wichtigsten drei Strukturvariablen sind `wType`, `iData` und `lData`. `wType` zeigt den Grund der Message an.

Signale sind Messages, die an ein Objekt gebunden sind. Wenn zum Beispiel ein Button gedrückt wurde, wird eine Message generiert, die sowohl den Grund des Auslösens, als auch das dazugehörige Objekt kodiert in `wType` enthält. Um dennoch Signale einfach auswerten zu können, gibt es das Makro „`SIGNAL`“.

```
SIGNAL(<ObjectID>, <PEG Status Flag>)
```

```
SIGNED MyWindow::Message(const PegMessage &Mesg) {  
    switch (Mesg.wType) {  
        case SIGNAL(IDB_EXE, PSF_CLICKED) :  
            //Code  
            break;  
        ...  
    }
```

Standard Button IDs:

IDB_ABORT = 1005	IDB_APPLY = 1004	IDB_CANCEL = 1003
IDB_CLOSE = 1000	IDB_COPY = 1025	IDB_EDIT_CLEAR_ALL = 1031
IDB_EDIT_COPY = 1027	IDB_EDIT_CUT = 1026	IDB_EDIT_DELETE = 1029
IDB_EDIT_PASTE = 1028	IDB_EDIT_REDO = 1033	IDB_EDIT_SELECT_ALL = 1030
IDB_EDIT_UNDO = 1032	IDB_ESC = 1021	IDB_EXE = 1020
IDB_FRAME_MENU = ???	IDB_GOTO = 1024	IDB_INPUT = 1022
IDB_NO = 1007	IDB_OK = 1002	IDB_RETRY = 1008
IDB_SET = 1023	IDB_SYSTEM = 1001	IDB_YES = 1006

PegThing Status Flags:

PSF_CHECK_OFF	PSF_CHECK_ON	PSF_CLICKED
PSF_CURRENT	PSF_DOT_OFF	PSF_DOT_ON
PSF_FOCUS_LOST	PSF_FOCUS_RECEIVED	PSF_KEY_RECEIVED
PSF_LIST_SELECT	PSF_RIGHTCLICK	PSF_SCROLL_CHANGE
PSF_SIZED	PSF_SLIDER_CHANGE	PSF_SPIN_LESS
PSF_SPIN_MORE	PSF_TEXT_EDIT	PSF_TEXT_EDIT_DONE
PSF_TEXT_SELECT	PSF_VISIBLE	PSF_PAGE_SELECT

Die folgenden Messages sind nicht an ein Objekt gebunden. Sie können direkt mit der Variable `wType` verglichen werden.

```

    SIGNED MyWindow::Message(const PegMessage &Mesg) {
        switch (Mesg.wType) {
            case PM_TIMER:
                //Code
                break;
            ...
        }
    }

```

PEG Messages:

PM_ADD	PM_ADDICON	PM_BEGIN_MOVE
PM_CLOSE	PM_CLOSE_SIBLINGS	PM_COPY
PM_CURRENT	PM_CUT	PM_DESTROY
PM_DIALOG_APPLY	PM_DIALOG_NOTIFY	PM_DRAW
PM_EXIT	PM_FIRST_START	PM_GAINING_FOCUS
PM_GET_INPUT_STATE	PM_HIDE	PM_HSCROLL
PM_KEY	PM_KEY_RELEASE	PM_LBUTTONDOWN
PM_LBUTTONUP	PM_LOSING_FOCUS	PM_MAXIMIZE
PM_MINIMIZE	PM_MOVE	PM_MOVE_FOCUS
PM_MWCOMPLETE	PM_NONCURRENT	PM_PARENTSIZED
PM_PASTE	PM_POINTER_ENTER	PM_POINTER_EXIT
PM_POINTER_MOVE	PM_RBUTTONDOWN	PM_RBUTTONUP
PM_REDRAW	PM_REMOVETHING	PM_RESTORE
PM_SHOW	PM_SIZE	PM_SLIDER_DRAG
PM_TIMER	PM_VSCROLL	

Frame Window Messages:

FWM_ACTIVATE_MAIN_WINDOW	FWM_ADD	FWM_CLOSE
FWM_FULL_SCREEN	FWM_KEYPAD_WAS_TOGGLED	FWM_MEMORY_ERROR_MESSAGE
FWM_QUERY_CLOSE	FWM_REMOVE	FWM_SELECT_ALL
FWM_SYS_CHANGED	FWM_UNDO	FWM_SPLIT_SCREEN

? Messages

MM_ABS	MM_DERIVATIVE	MM_EXP
MM_INTEGRAL	MM_LIMIT	MM_LINEBREAK
MM_LIST	MM_LOG	MM_MATRIX_HORIZONTAL
MM_MATRIX_SQUARE	MM_MATRIX_VERTICAL	MM_PARENTHESES
MM_POWER	MM_PRODUCT	MM_ROOT
MM_SIMULTANEOUS	MM_SQUARE_ROOT	MM_SUM
MM_SELECTION_OVER_EMPTY		

? Messages

UM_ACTIVATE_UNDO	UM_CHILD_CURRENT	UM_CHILD_NONCURRENT
UM_CHILD_RESIZED	UM_CONTENTS_CHANGED	UM_CURSOR_MOVED
UM_GET_DRAG_OBJECT	UM_SELECTION_ENTER	UM_SELECTION_EXIT
UM_SELECTION_MOVED	UM_DRAG_ENTER	

? Messages

HM_SYS_CLEAR	HM_SYS_CLOSE	HM_SYS_CONTINUE
HM_SYS_KEYBOARD	HM_SYS_OPTION_MENU	HM_SYS_RESUME
HM_SYS_SWAP	HM_SYS_ZOOM	HM_SYS_TEST

PROBLEME WÄHREND DER ENTWICKLUNG

Im Laufe des Projektes sind wir von einigen Eigenheiten in der Classpad-Add-in-Entwicklung überrascht worden. Der Taschenrechner hatte die Betriebssystemversion 3.05.

- Einige aus dem C++-Umfeld standardisierte Klassen waren im Classpad-SDK nicht verfügbar. Als Beispiel ist die Klasse List zu nennen. Möglicherweise ersetzt die Klasse CPList die Standardklasse, sie wurde aber von uns nicht getestet. Wir haben als Alternative eine eigene Listenimplementation verwendet.
- Beim Zeichnen von Elementen außerhalb des momentanen Sichtbereichs wurden die Koordinaten der Eckpunkte verändert.
- Der Taschenrechner und der PC-Emulator weisen ein unterschiedliches Verhalten auf. Zum Beispiel bei der Anzeige für unendlich. Der Emulator zeigt „1.#INF“, der Classpad zeigt „+++++“.

ANHANG

A: KLASSENLISTE (UNVOLLSTÄDIG)

Klassenname	Konstruktor
PegBitmapButton	<ul style="list-style-type: none"> • PegRect &rect, PegBitmap *bm, WORD wld=0, WORD wStyle=AF_ENABLED, UCHAR type=TYPE_BMBUTTON, BOOL blInvertOnSelect=TRUE • SIGNED iLeft, SIGNED iTop, PegBitmap *bm, WORD wld=0, WORD wStyle=AF_ENABLED, • BOOL blInvertOnSelect=TRUE
PegButton	<ul style="list-style-type: none"> • const PegRect &rect, WORD wld=0, WORD wFlags=AF_ENABLED • WORD wld=0, WORD wFlags=AF_ENABLED
PegCapture	<ul style="list-style-type: none"> • void
PegCheckBox	<ul style="list-style-type: none"> • PegRect &rect, const PEGCHAR *text, WORD wld=0, WORD Style=AF_ENABLED • SIGNED iLeft, SIGNED iTop, const PEGCHAR *text, WORD wld=0, WORD wStyle=AF_ENABLED
PegColor	<ul style="list-style-type: none"> • void • COLORVAL fore, COLORVAL back=WHITE, UCHAR Flags=CF_NONE
PegComboBox	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld=0, WORD wStyle=FF_THIN
PegContainerVisitor	<ul style="list-style-type: none"> • ???
PegDecoratedWindow	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wStyle=FF_THICK • WORD wStyle=FF_THICK
PegDialog	<ul style="list-style-type: none"> • const PegRect &Rect, const PEGCHAR *Title=NULL, PegThing *ReportTo=NULL, WORD wStyle=FF_RAISED • const PEGCHAR *Title=NULL, PegThing *ReportTo=NULL, WORD wStyle=FF_RAISED
PegDoDestroyMessageVisitor	<ul style="list-style-type: none"> • ???
PegDoNotDestroyVisitor	<ul style="list-style-type: none"> • PegThing *t
PegEditBox	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld = 0, WORD wStyle = FF_RECESSED EF_EDIT EF_WRAP, PEGCHAR *Text = NULL, WORD wMaxChars = 1000
PegEditBoxInputState	<ul style="list-style-type: none"> • ???

PegExpString	<ul style="list-style-type: none"> • ???
PegFixedWidthButton	<ul style="list-style-type: none"> • SIGNED wLeft, SIGNED wTop, WORD cmnID • SIGNED wLeft, SIGNED wTop, WORD cmnID, WORD outputID
PegFoldMessageVisitor	<ul style="list-style-type: none"> • ???
PegHorzList	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld = 0, WORD wStyle = FF_THIN
PegHScroll	<ul style="list-style-type: none"> • const PegRect &InRect, PegScrollInfo *si, WORD wld = 0
PegInputCursor	<ul style="list-style-type: none"> • ???
PegInputState	<ul style="list-style-type: none"> • ???
PegList	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld = 0, WORD wStyle = FF_THIN
PegMenu	<ul style="list-style-type: none"> • PegMenuDescription *pDesc, BOOL bPopup=FALSE • PegMenuDescriptionML *pDesc, BOOL bPopup=FALSE, UCHAR ucMethod=MENU_METHOD_STATIC
PegMenuBar	<ul style="list-style-type: none"> • PegMenuDescription *pDesc
PegMenuButton	<ul style="list-style-type: none"> • PegMenuDescription *pDesc
PegMessage	<ul style="list-style-type: none"> • void • WORD wVal • PegThing *pTo, WORD wVal
PegMessageQueue	<ul style="list-style-type: none"> • void
PegMessageTypeVisitor	<ul style="list-style-type: none"> • WORD w
PegMessageVisitor	<ul style="list-style-type: none"> • ???
PegMessageWindow	<ul style="list-style-type: none"> • const PegRect &Rect, const PEGCHAR *Title, const PEGCHAR *Message = NULL, WORD wStyle = MW_OK FF_RAISED, PegBitmap *plcon = NULL, PegThing *Owner = NULL • const PEGCHAR *Title, const PEGCHAR *Message = NULL, WORD wStyle = MW_OK FF_RAISED, PegBitmap *plcon = NULL, PegThing *Owner = NULL
PegNotebook	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wStyle, UCHAR uNumTabs
PegPresentationManager	<ul style="list-style-type: none"> • PegRect &Size
PegProgressBar	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wStyle = FF_THIN PS_SHOW_VAL PS_PERCENT, SIGNED iMin = 0, SIGNED iMax = 100, SIGNED iCurrent = 0
PegProgressWindow	<ul style="list-style-type: none"> • const PegRect &Rect, const PEGCHAR *Title, const PEGCHAR *Message,

	<p>WORD wMesgStyle, WORD wProgStyle = FF_THIN PS_SHOW_VAL PS_PERCENT, PegBitmap *plcon = NULL, PegThing *Owner = NULL</p> <ul style="list-style-type: none"> const PEGCHAR *Title, const PEGCHAR *Message, WORD wMesgStyle, WORD wProgStyle = FF_THIN PS_SHOW_VAL PS_PERCENT, PegBitmap *plcon = NULL, PegThing *Owner = NULL
PegPrompt	<ul style="list-style-type: none"> const PegRect &Rect, const PEGCHAR *Text, WORD wld = 0, WORD wStyle = FF_NONE TJ_LEFT AF_TRANSPARENT SIGNED iLeft, SIGNED iTop, SIGNED iWidth, const PEGCHAR *Text, WORD wld = 0, WORD wStyle = FF_NONE TJ_LEFT AF_TRANSPARENT, PegFont *pFont = NULL SIGNED iLeft, SIGNED iTop, const PEGCHAR *Text, WORD wld = 0, WORD wStyle = FF_NONE TJ_LEFT AF_TRANSPARENT, PegFont *pFont = NULL
PegRadioButton	<ul style="list-style-type: none"> PegRect &Rect, const PEGCHAR *text, WORD wld = 0, WORD wStyle = AF_ENABLED SIGNED iLeft, SIGNED itop, const PEGCHAR *text, WORD wld = 0, WORD wStyle = AF_ENABLED
PegScreen	<ul style="list-style-type: none"> const PegRect &Size
PegScrollbar	<ul style="list-style-type: none"> BOOL bAmVertical
PegSourceVisitor	<ul style="list-style-type: none"> PegThing *
PegSpinButton	<ul style="list-style-type: none"> SIGNED iLeft, SIGNED iTop, WORD wld = 0, WORD wStyle = AF_ENABLED SB_VERTICAL const PegRect &Rect, WORD wld = 0, WORD wStyle = AF_ENABLED SB_VERTICAL SIGNED iLeft, SIGNED iTop, PegTextThing *buddy, LONG IMin, LONG IMax, SIGNED iStep, WORD wld = 0, WORD wStyle = AF_ENABLED SB_VERTICAL const PegRect &Rect, PegTextThing *buddy, LONG IMin, LONG IMax, SIGNED iStep, WORD wld = 0, WORD wStyle = AF_ENABLED SB_VERTICAL

PegStatusBar	<ul style="list-style-type: none"> • Void
PegString	<ul style="list-style-type: none"> • const PegRect &Rect, const PEGCHAR *Text = NULL, WORD wld = 0, WORD wStyle = FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen = -1 • SIGNED iLeft, SIGNED iTop, SIGNED iWidth, const PEGCHAR *Text = NULL, WORD wld = 0, WORD wStyle = FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen = -1 • SIGNED iLeft, SIGNED iTop, const PEGCHAR *Text = NULL, WORD wld = 0, WORD wStyle = FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen = -1
PegStringInputState	<ul style="list-style-type: none"> • ???
PegSystemButton	<ul style="list-style-type: none"> • ???
PegTextBox	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld = 0, WORD wStyle = FF_RECESSED EF_WRAP TJ_LEFT, PEGCHAR *Text = NULL, WORD wMaxChars = 1000
PegTextButton	<ul style="list-style-type: none"> • PegRect &Rect, const PEGCHAR *text, WORD wld = 0, WORD wStyle = AF_ENABLED • SIGNED iLeft, SIGNED iTop, SIGNED iWidth, const PEGCHAR *text, WORD wld = 0, WORD wStyle = AF_ENABLED • SIGNED iLeft, SIGNED iTop, const PEGCHAR *text, WORD wld = 0, WORD wStyle = AF_ENABLED
PegTextInputCursor	<ul style="list-style-type: none"> • ???
PegTextThing	<ul style="list-style-type: none"> • const PEGCHAR *Text, WORD wCopy = 0 • WORD wCopy = 0
PegThing	<ul style="list-style-type: none"> • onst PegRect &Rect, WORD wld = 0, WORD wStyle = FF_NONE • WORD wld = 0, WORD wStyle = FF_NONE • ...
PegThingVisitor	<ul style="list-style-type: none"> • ???
PegThinMenuButton	<ul style="list-style-type: none"> • ???
PegTitle	<ul style="list-style-type: none"> • const PEGCHAR *Title, WORD wStyle = TF_SYSBUTTON TF_MINMAXBUTTON TF_CLOSEBUTTON
PegToolBar	<ul style="list-style-type: none"> • WORD wID=0
PegToolBarPanel	<ul style="list-style-type: none"> • WORD wID=0

PegVertList	<ul style="list-style-type: none"> const PegRect &Rect, WORD wld = 0, WORD wStyle = FF_THIN
PegVScroll	<ul style="list-style-type: none"> const PegRect &InRect, PegScrollInfo *si, WORD wld = 0 void
PegWindow	<ul style="list-style-type: none"> const PegRect &Rect, WORD wStyle = FF_THICK WORD wStyle = FF_THICK

B: CLASSPAD SPEZIFISCHE KLASSEN

Klassenname	Konstruktor	Beschreibung
CPApp		
CPArray	<ul style="list-style-type: none"> void 	Dynamisches Pointerarray
CPBitFieldReadBuffer		
CPBitFieldWriteBuffer		
CPBitmapButton	<ul style="list-style-type: none"> SIGNED iLeft, SIGNED iTop, PegBitmap *bm, WORD wld=0, WORD wStyle=AF_ENABLED, BOOL blInvertOnSelect=TRUE 	
CPButtonGrid		
CPCapsLockKey		
CPClipboard	<ul style="list-style-type: none"> void 	
CPComboBox		
CPCompactRect	<ul style="list-style-type: none"> void PegRect rect CPoint top_left, CPoint bottom_right short x0, short y0, short x1, short y1 	
CPDialog	<ul style="list-style-type: none"> const PegRect &Rect, const PEGCHAR *Title=NULL, PegThing *ReportTo=NULL, WORD wStyle=FF_RAISED, KeypadState show_keypad=CP_KEYPAD_OFF const PEGCHAR *Title=NULL, PegThing *ReportTo=NULL, WORD wStyle=FF_RAISED, KeypadState show_keypad=CP_KEYPAD_OFF const DialogSize size, const DialogPosition position, const PEGCHAR *Title=NULL, PegThing *ReportTo=NULL, WORD wStyle=FF_RAISED, KeypadState show_keypad=CP_KEYPAD_OFF 	
CPDocument	<ul style="list-style-type: none"> CPMainFrame *mainframe 	Datenaustausch zwischen Fenstern

CPDragAndDropObject	<ul style="list-style-type: none"> • void 	
CPDropDownButton	<ul style="list-style-type: none"> • PegRect &Rect, const CPMultiButtonDescription *desc, char cols=1, char size=1, BOOL as_cols=TRUE, WORD wStyle=AF_ENABLED 	
CPEditBox	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wld=0, WORD wStyle=FF_RECESSED EF_EDIT EF_WRAP , PEGCHAR *Text=NULL, WORD wMaxChars=1000 	
CPEXpression	<ul style="list-style-type: none"> • void • const char *s, BOOL eval=FALSE • const CPString &s, BOOL eval=FALSE • const CPEXpression &x • const OBCD &x • const int i 	String -> OBCD
CPFile	<ul style="list-style-type: none"> • void 	
CPFileList		
CPFileSystem		
CPFileWindow		
CPFlagReader		
CPFlagWriter		
CPFlashFileSystem		
CPFrameWindow	<ul style="list-style-type: none"> • const PegRect &Rect, WORD wStyle=FF_NONE • WORD wStyle=FF_NONE 	
CPFuzzyExpression		
CPHotLinkClient		
CPHotLinkMathClient		
CPHotLinkObject		
CPHotLinkServer		
CPHotLinkTextClient		
CPKey		
CPKeyboard		
CPKeyboardManager		
CPKeyGrid		
CPKeyGridBitmapKeyContents		
CPKeyGridButtonKeyContents		
CPKeyGridCharContents		
CPKeyGridContents		
CPKeyGridContentsSender		
CPKeyGridStringContents		
CPKeyLayoutBar		
CPKeyLayoutBarParent		
CPKeypad		
CPLayout		
CPLengthDocument		
CPLine	<ul style="list-style-type: none"> • void • CPoint p, CPoint q 	
CPLineEdit		
CPLineItem		

CPLinkableObject		
CPList	<ul style="list-style-type: none"> • void • ent a 	
CPListBox		
CPListBoxConstantContents		
CPListBoxContents		
CPListKey		
CPListIterator	<ul style="list-style-type: none"> • const CPList *s 	
CPMainFrame	<ul style="list-style-type: none"> • PegRect &Rect, int frame_ui_width=DEFAULT_FRAME_UI_WIDTH 	
CPMathInputClipboard	<ul style="list-style-type: none"> • CLinearMathObject &ob 	
CPMCSFileSystem		
CPMEMFileHeader	<ul style="list-style-type: none"> • const PEGCHAR *AppName, const PEGCHAR *DataName, PEGCHAR MajorVersion=1, PEGCHAR MinorVersion=0 	
CPModuleWindow	<ul style="list-style-type: none"> • PegRect rect, CPModuleWindow *invoking_window, CPDocument *doc, CPMainFrame *frame 	
CPMultiButton	<ul style="list-style-type: none"> • PegRect &Rect, const CPMultiButtonDescription *desc, WORD wStyle=AF_ENABLED 	
CPObjectFactory		
CPPegString	<ul style="list-style-type: none"> • const PegRect &Rect, const PEGCHAR *Text=NULL, WORD wld=0, WORD wStyle=FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen=DEFAULT_PEG_STRING_BYTE_LIMIT • SIGNED iLeft, SIGNED iTop, SIGNED iWidth, const PEGCHAR *Text=NULL, WORD wld=0, WORD wStyle=FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen=DEFAULT_PEG_STRING_BYTE_LIMIT • SIGNED iLeft, SIGNED iTop, const PEGCHAR *Text=NULL, WORD wld=0, WORD wStyle=FF_RECESSED AF_ENABLED EF_EDIT, SIGNED iLen=DEFAULT_PEG_STRING_BYTE_LIMIT 	
CPPoint	<ul style="list-style-type: none"> • void • short xx, short yy 	
CPReadFile	<ul style="list-style-type: none"> • void 	
CPReadFlashFile		

CPReadMCSFile	<ul style="list-style-type: none"> • UCHAR type • const char *name, const char *path=NULL, UCHAR type=0 	
CPRect	<ul style="list-style-type: none"> • void • PegRect rect 	
CPRectWithBaseline		
CPRolloverButton		
CPSequenceableObject		
CPShiftKey		
CPShiftTextKey		
CPString	<ul style="list-style-type: none"> • CPMCHAR c • const PEGCHAR *s • const CPString &y • OBCD d, int dummy=0 	Zeichenkette
CPStrings		
CPTabbedWindow	<ul style="list-style-type: none"> • PegRect rect 	
CPTextClipboard	<ul style="list-style-type: none"> • const PEGCHAR *str • const CPString &str 	
CPTextKey		
CPToggleButton	<ul style="list-style-type: none"> • PegRect &Rect, PegBitmap *bm, WORD wld=0, WORD wStyle=AF_ENABLED 	
CPUndoGroup		
CPUndoThing		
CPWindow		
CPWriteFile	<ul style="list-style-type: none"> • void • const char *name, const char *path=NULL 	
CPWriteFlashFile		
CPWriteMCSFile	<ul style="list-style-type: none"> • UCHAR type • const char *name, const char *path=NULL, UCHAR type=0 	

C: SONSTIGE KLASSEN

Klassenname	Konstruktor	Beschreibung
InputDialog	<ul style="list-style-type: none"> • PEGCHAR *title, PEGCHAR *prompt1, OBCD &data1, PEGCHAR *prompt2, OBCD &data2, BOOL bDisp=FALSE, BOOL bTab=FALSE, SIGNED iData=0 • (PEGCHAR *title, PEGCHAR *prompt, OBCD &data, SIGNED key=0 • PEGCHAR *title, PEGCHAR *prompt, CP_CHAR *pBuff, SIGNED iLen=-1 	
MathWindow	<ul style="list-style-type: none"> • PegRect &rect, 	

	bool size_to_fit=false, WORD wld=0, bool read_only=false, SIGNED maxObjects=DEFAULT_MAX_OBJECTS, SIGNED maxChars=DEFAULT_MAX_CHARS	
SCEvent	<ul style="list-style-type: none"> • EventType type, CPoint x, unsigned int extra_info=0 	
SCMode	<ul style="list-style-type: none"> • void 	
ScrollableMathWindow	<ul style="list-style-type: none"> • PegRect rect, WORD wld=0, WORD wStyle=FF_THIN, bool read_only=false 	
SCWindow	<ul style="list-style-type: none"> • const PegRect &rect, WORD wStyle=FF_NONE • WORD wStyle=FF_NONE 	
SCWindowWithMode	<ul style="list-style-type: none"> • PegRect &rect 	
SlidingMathWindow	<ul style="list-style-type: none"> • PegRect rect, WORD wld=0, WORD wStyle=FF_THIN, bool read_only=false, SIGNED maxObjects=DEFAULT_MAX_OBJECTS, SIGNED maxChars=DEFAULT_MAX_CHARS 	
StorageManager	<ul style="list-style-type: none"> • CPString *varname, CPString *pathname, ActionStates action, UCHAR type=0 	
StoreDialog	<ul style="list-style-type: none"> • PEGCHAR *title, CP_CHAR *folder, CP_CHAR *file, UCHAR type, BOOL bRecall • PEGCHAR *title, CP_CHAR *folder, CP_CHAR *file, UCHAR type 	
TabArrowMathWindow	<ul style="list-style-type: none"> • PegRect rect, int math_style=MS_LEFT_JUSTIFIED, WORD wld=0, WORD wStyle=FF_NONE, bool read_only=false, SIGNED maxObjects=DEFAULT_MAX_OBJECTS, SIGNED maxChars=DEFAULT_MAX_CHARS 	
TextMathWindow	<ul style="list-style-type: none"> • PegRect &rect, • bool size_to_fit=false, WORD wld=0, bool read_only=false 	
ValidFolder	<ul style="list-style-type: none"> • const PEGCHAR *name • VALMANSTRUCT *pValManAdr 	

QUELLEN

<http://www.casiocalc.org/index.php?showtopic=2839&view=findpost&p=35899> Zugriff am: 26.02.2011

https://dip.felk.cvut.cz/browse/pdfcache/vankom2_2008bach.pdf Zugriff am: 10.02.2011

http://documentation.renesas.com/eng/products/mpumcu/rej09b0081_sh7709s.pdf Zugriff am:
26.02.2011

<http://kurser.iha.dk/eit/embedded/OnTime/peg-manualer/pegman/refman.htm> Zugriff am: 26.02.2011

CASIO ClassPad 300 SDK Installation Guide

CASIO ClassPad 300 SDK Programing Guide

CASIO ClassPad 300 SDK Reference

ClassPad SDK (Programing) Tutorial