

# Bibliotheken (SDL)

## SVN

## IDEs



Jakob Karge

# Inhalt

## 1 Bibliotheken

1.1 Standard C-Library

1.2 SDL

## 2 SVN

## 3 IDEs

## 4 Literatur



# 1 Bibliotheken

- **Sammlung von Programmfunktionen**
- **Lösung wiederkehrender Probleme**
- **Abstraktion von Betriebssystem- und Hardwareprogrammierung**
- **Vorteile:**
  - **Zeitersparnis**
  - **Fehlerreduktion**
  - **unabhängige Pflege**
  - **Portabilität**



# 1 Bibliotheken

- **Statisch**

- statische Bibliotheken werden an das Programm gelinkt
- hängen an jedem Programm das sie benutzt mit dran
- Dateityp vor dem Linken: `.a` / `.lib`

- **Dynamisch**

- dynamische Bibliotheken liegen als separate Dateien vor (an fest definierten Orten)
- (bestenfalls) nur einmalig pro System vorhanden
- Dateitypen: `.so` / `.dll` / `.dylib`



# 1.1 Standard C Library

- **Kein fester Bestandteil der Sprache C**
- **Implementierung Compilerabhaengig**
  - Linux: glibc, uClipc, dietlibc
  - Windows: MS Visual C++ Laufzeitumgebung msvcrt.dll
- **C99 Library umfasst 482 Funktionen**
  - Java SE6 API hat über 21.000 Funktionen
- **Nutzen durch Einbinden der Headerdateien**



# 1.1 Standard C Library

- **Inhalt (Auszug)**

- **stdio.h - Ein- und Ausgabe**

- `printf()`, `scanf()`, `fgets()`, `fopen()`

- **string.h - Manipulation von Zeichenketten**

- `strncpy()`, `strlen()`, `strcmp()`, `memcpy()`

- **stdlib.h - Speicherverwaltung, Zahlenkonvertierung**

- `malloc()`, `atoi()`, `rand()`, `qsort()`

- **math.h - Mathematische Funktionen**

- `sin()`, `cos()`, `pow()`, `log()`, `sqrt()`

- **stdbool.h - definiert den Datentyp `bool`**

- **complex.h - Operationen für komplexe Zahlen**







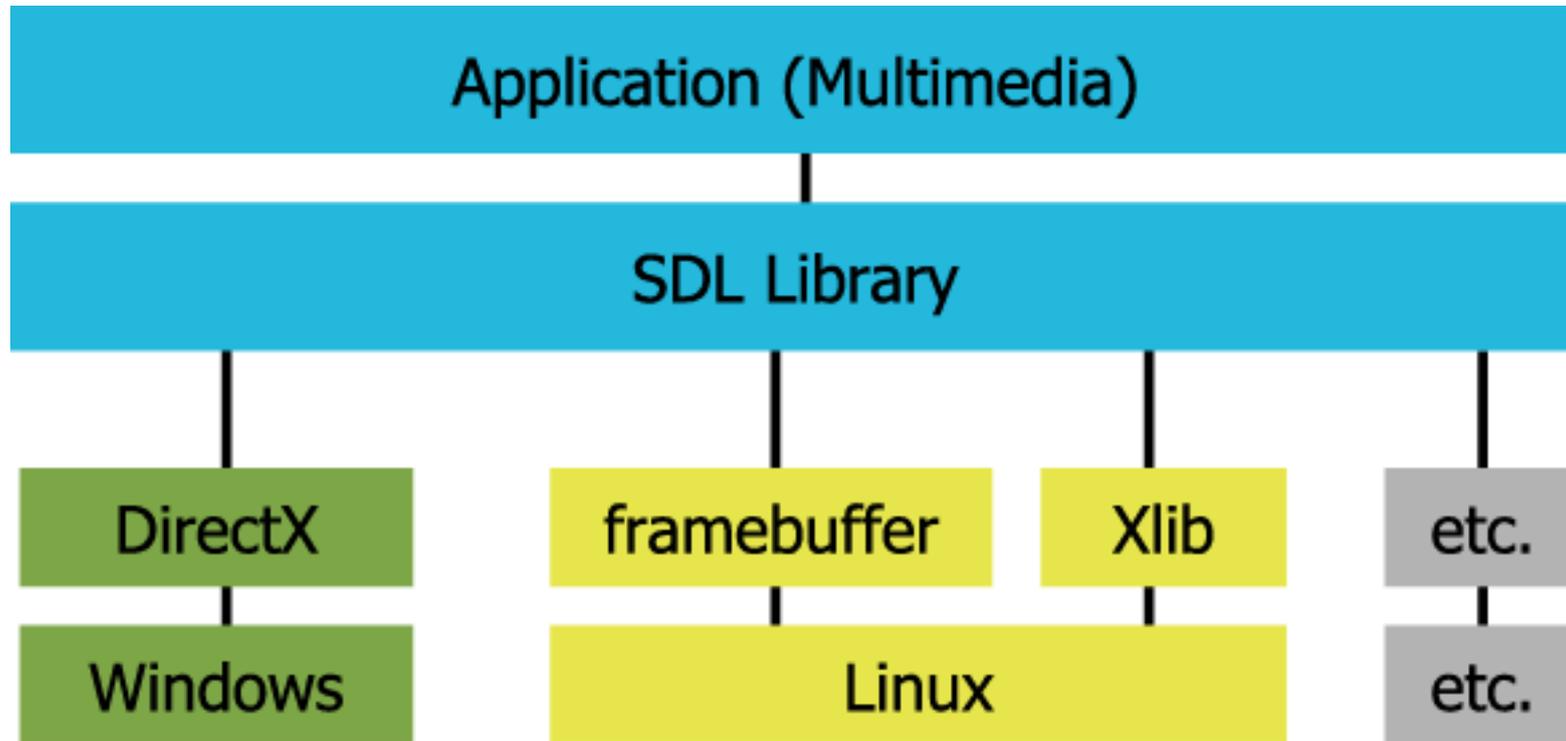
## 1.2 SDL

- **Simple Directmedia Layer**
- **Plattformunabhängige Medienbibliothek**  
(Linux, Windows, MacOS, BSD, ...)
- **Low-Level-Zugriff auf 2D/3D Video-, Audio-, I/O-Schnittstellen, Threading, ...**
- **Freie und Open-Source Software (LGPL)**

[www.libsdl.org](http://www.libsdl.org)



## 1.2 SDL



- **Dokumentation:** [http://www.libsdl.org/cgi/docwiki.cgi/SDL\\_API](http://www.libsdl.org/cgi/docwiki.cgi/SDL_API)
- **Tutorials:** [http://lazyfoo.net/SDL\\_tutorials](http://lazyfoo.net/SDL_tutorials)
- **Links:** <http://www.libsdl.de/docs.htm>



# 1.2 SDL

Live Demo: TicTacToe



## 2 SVN

- Apache Subversion
- **Versionskontrolle**
  - verwaltet Änderungen in Dateien
  - jede Version kann jederzeit wiederhergestellt werden
  - mehrere Benutzer können gleichzeitig zugreifen
  - **Arbeiten auf einer lokalen Kopie des Repositorys**  
checkout, update
  - **Veränderungen werden zurück ins Repository geschrieben**  
commit / checkin
  - **Abzweigen einer separat verwalteten Variante möglich**  
branch



# 2 SVN

- **Die wichtigsten Befehle**

- **Neue Repositorykopie anlegen**

```
svn checkout [repolink]
```

- **Updates vom Repository in die lokale Kopie laden**

```
svn up / svn update
```

- **Änderungen ins Repository schreiben**

```
svn ci / svn commit
```

- **Eine neue Datei zur Versionsverwaltung hinzufügen**

```
svn add [file]
```

- **lokale Änderungen verwerfen und zur Version V zurückkehren**

```
svn revert V
```



## 2 SVN

- **Warum erzählen wir euch von SVN?**
  - Versionskontrolle hilft bei der Gruppenarbeit an Code
  - Fehler können leicht revidiert werden
  - Ihr braucht SVN in TechGl3 um die HA abzugeben
- **Es gibt noch mehr:**
  - CVS ist der Vorläufer von SVN und inzwischen veraltet
  - Git und Mercurial sind moderne sog. „verteilte“ Versionsverwaltungssysteme
  - verteilte Versionskontrolle hat keinen zentralen Server

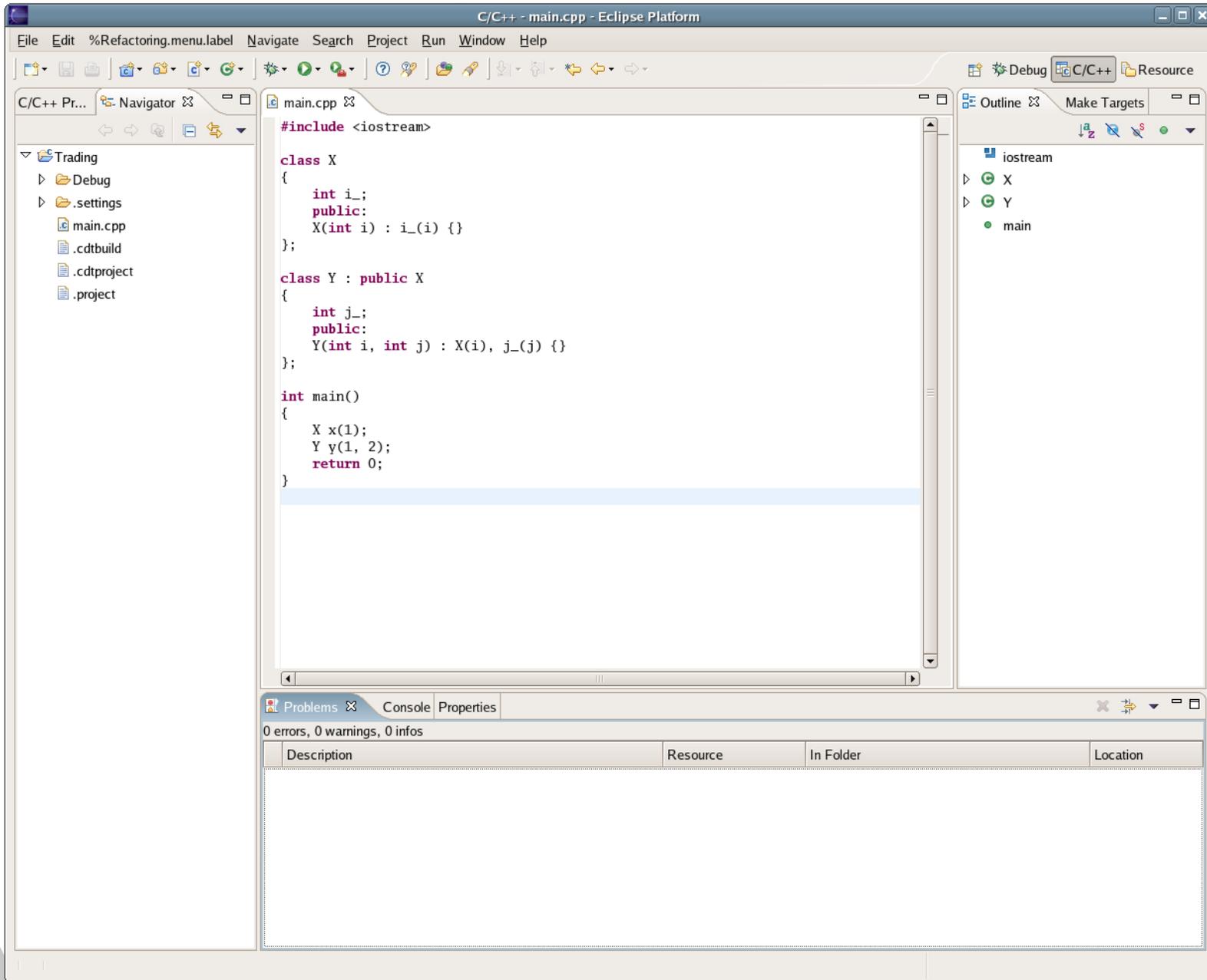


# 3 IDE

- Integrated Development Environment
- **verbinden Editor, Compiler, Debugger, Versionsverwaltung in einer Umgebung**
- **bieten Code- & Sprach-spezifische Features**
  - Syntax-Highlighting
  - Autovervollständigung
  - Projektstrukturierung
  - Templates
  - ...

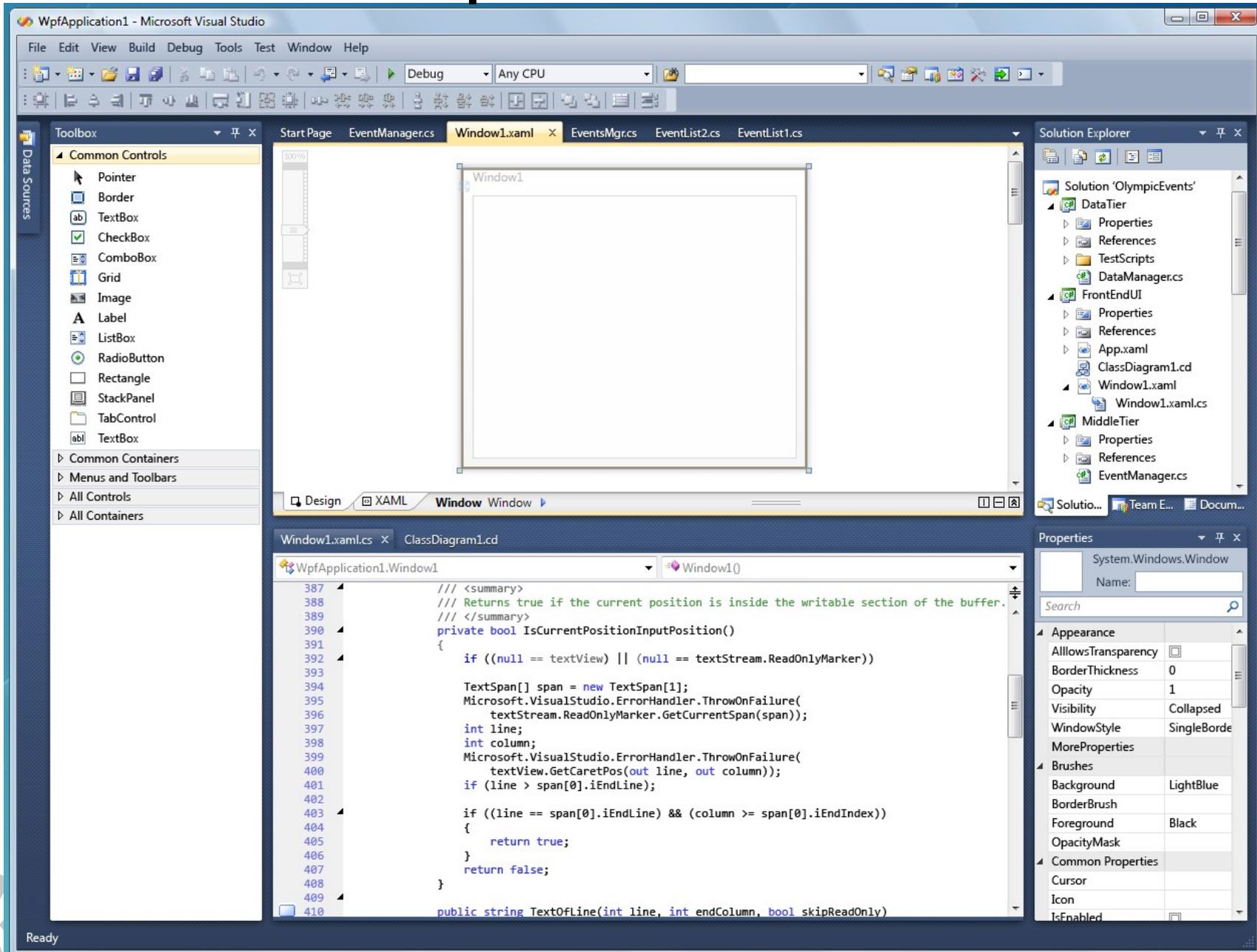


# Beispiele für IDEs



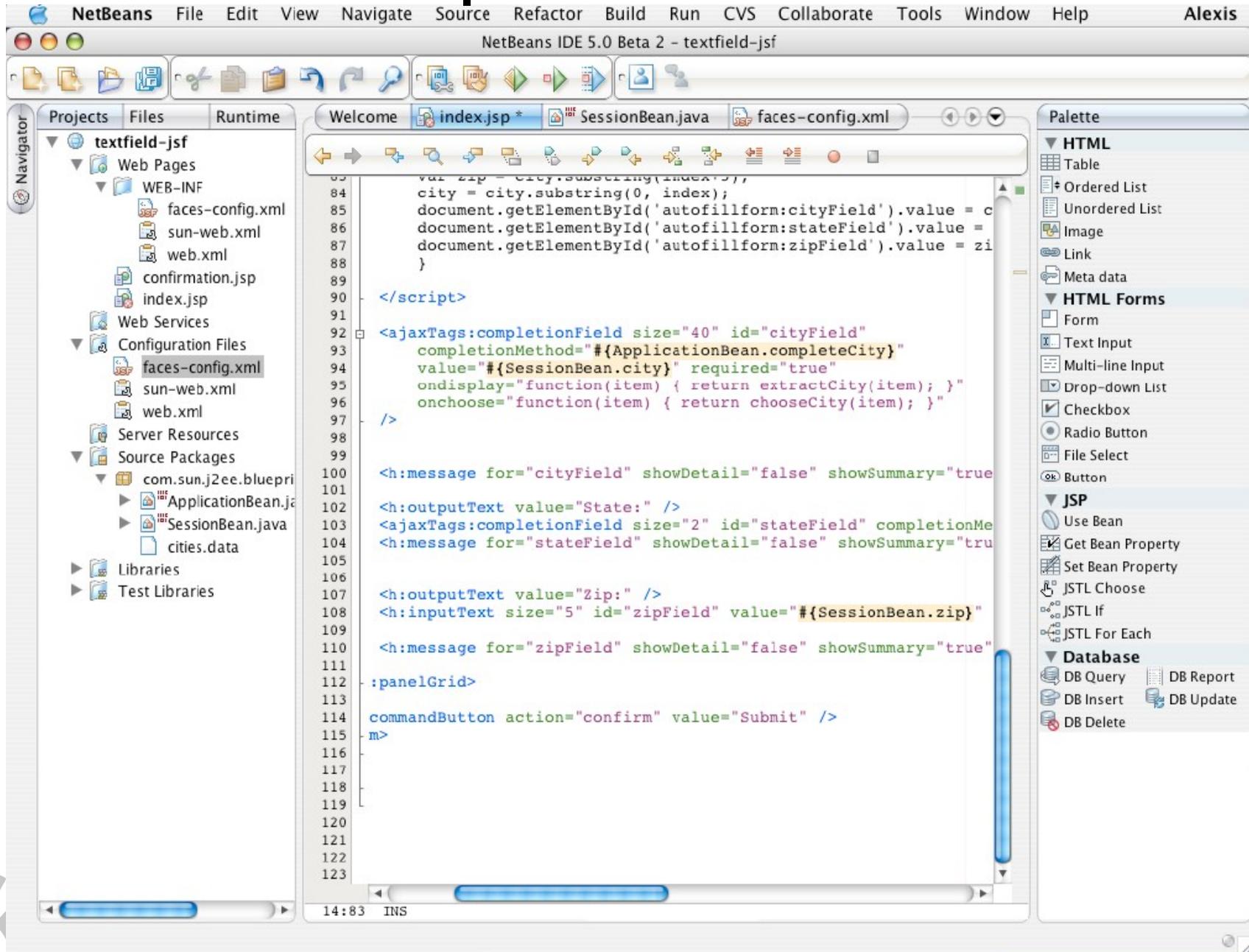
Eclipse CDT (Linux, Windows, MacOS)

# Beispiele fuer IDEs



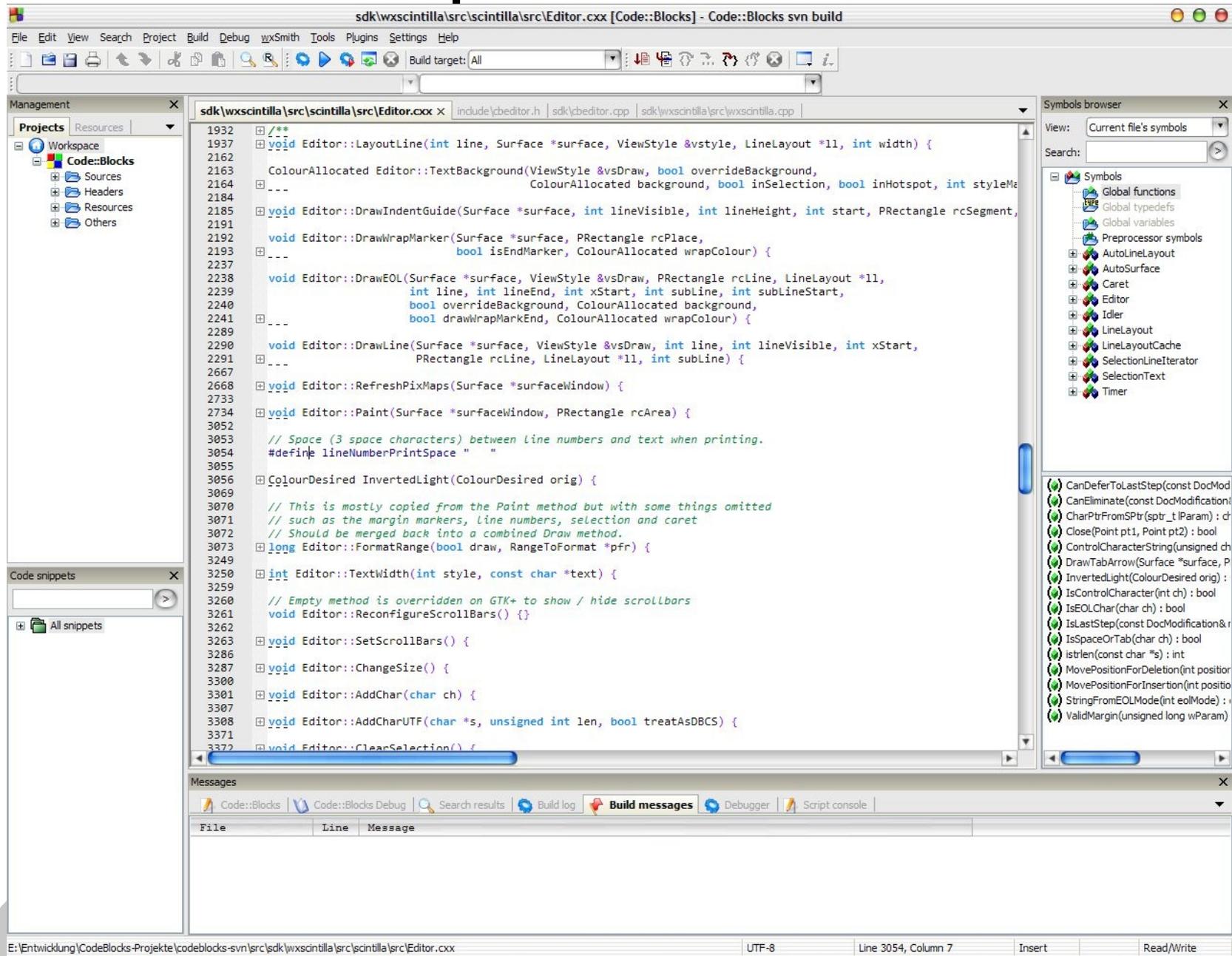
Visual Studio / Visual C++ (Windows)

# Beispiele fuer IDEs



NetBeans (Linux, Windows, MacOS)

# Beispiele fuer IDEs



Code::Blocks (Linux, Windows, MacOS)

# 4 Links und Literatur

- **Bruce Eckel - „Thinking in C - A Flash Based Audio-Visual Seminar“**

<http://www.mindviewinc.com/CDs/ThinkingInC>

- Flash-basierter Multimedia-Kurs (englisch)

- **Prof. Dr. Peter Baeumle-Courth - „Einführung in ANSI-C“**

<http://info.baeumle.com/ansic.html>

- An Kernighan und Richie angelehntes Script
- Übersichtlich und gut strukturiert

- **Wikibooks: „C-Programmierung Einsteigerkurs“**

<http://de.wikibooks.org/wiki/C-Programmierung>

- Kollaboratives Werk
- In verschiedenen Sprachen und als PDF erhältlich



# 4 Links und Literatur

- **UNIX manual-Pages**

- Suche nach Stichworten via apropos-Kommando, z.B.:

```
apropos printf
```

- Zugriff auf manual-Pages von System- und Bibliotheksaufrufen unter UNIX-Systemen via man-Kommando:

```
man -s 3c printf
```

- **Online verfügbare man-Archive:**

- **Man-Pages der Debian-Linux-Distribution:**

<http://manpages.debian.net>

- **Umfangreiches man-Page-Archiv:**

<http://linuxmanpages.com>



# 4 Links und Literatur

- **C Standard Library**

Online-Referenz der Standard C Library unter:

<http://www.cplusplus.com/reference/clibrary>

- **Falls euch C langweilt, lernt C++, denn:**

- C++ erlaubt objektorientiertes Programmieren
- C++ ist performant
- Es gibt umfangreiche Bibliotheken für C++
- Einblicke in (in Java verborgene) Details
- **Kostenloses ebook: Bruce Eckel - Thinking in C++**

<http://www.mindview.net/Books/TICPP/ThinkingInCPP2e.html>



# 4 Links und Literatur

- **Helmut Erlenkötter - „C Programmieren von Anfang an“**

(2. Auflage 1999; Rowohlt Taschenbuch; ISBN 3499600749)

- Leicht verständlich, für Programmierneinsteiger geeignet
- Viele Beispiele, Aufgaben am Kapitelende
- 1. Platz der Amazon-Verkaufs-Charts

- **Peter van der Linden - „Expert C Programming“**

(1. Edition 1994; Prentice Hall; ISBN 0131774298)

- Detailliertes Hintergrundwissen, best practises
- Humorvoller Stil, viele Anekdoten
- Für erfahrene C Programmierer geeignet
- Nur auf Englisch erhältlich



# 4 Links und Literatur

- **B. W. Kernighan, D. M. Ritchie - „The C Programming Language“**  
(2. Edition 1988; Prentice Hall; ISBN 0131103628)
  - DAS Standardwerk, gut als Nachschlagewerk geeignet
  - Für Programmieranfänger weniger geeignet
  - Letzte Auflage von von 1988, kein C99
  - 274 Seiten, enthält Referenz der C Standardbibliothek
- **Peter Prinz, Tony Crawford - „C in a Nutshell“**  
(1. Auflage 2006; O'Reilly; ISBN 3897213443)
  - Sehr aktuell, basiert auf ISO/IEC 9899:1999 (C99)
  - Umfangreich auf über 600 Seiten
  - Umfasst C, Standardbibliothek, GNU-Tools (gcc, gdb, make)
  - Nicht unbedingt für Einsteiger geeignet



# 4 Links und Literatur

- **Peter Prinz, Ulla Kirch-Prinz - „C - Kurz & gut“**  
(1. Auflage 2002; O'Reilly; ISBN 3897212382)
  - Nachschlagewerk für Sprache C und Standardbibliothek
  - Preiswert erhältlich
  
- **Arnold Robbins - „GDB - Kurz & gut“**  
(1. Auflage 2005; O'Reilly; ISBN 3897215128)
  - Nachschlagewerk für den GNU Debugger (GDB)
  - Preiswert erhältlich



# Quellen

**Originalfolien von Martin Kresse (2010)**

**Bilder Abschnitt SDL ebenda**

**Bilder IDEs:**

<http://www.weiqigao.com/blog/images/eclipse-cdt.png>

[http://www.sdtimes.com/blog/image.axd?picture=2009%2F2%2FDvX\\_ShellBase.png](http://www.sdtimes.com/blog/image.axd?picture=2009%2F2%2FDvX_ShellBase.png)

[http://netbeans.org/images\\_www/screenshots/5.0/NetBeans50beta2Mac.png](http://netbeans.org/images_www/screenshots/5.0/NetBeans50beta2Mac.png)

<http://forums.codeblocks.org/index.php/topic,5426.0>

