به نام خدا

سلسله کارگاه های برنامه سازی تخصصی و کاربردی (کارگاه اول)

برنامه نویسی واسط کاربری گرافیکی

GUI (Graphical User Interface) Programming

کارشناس مهندسی نرم افزار

ارائه: عباس نادری

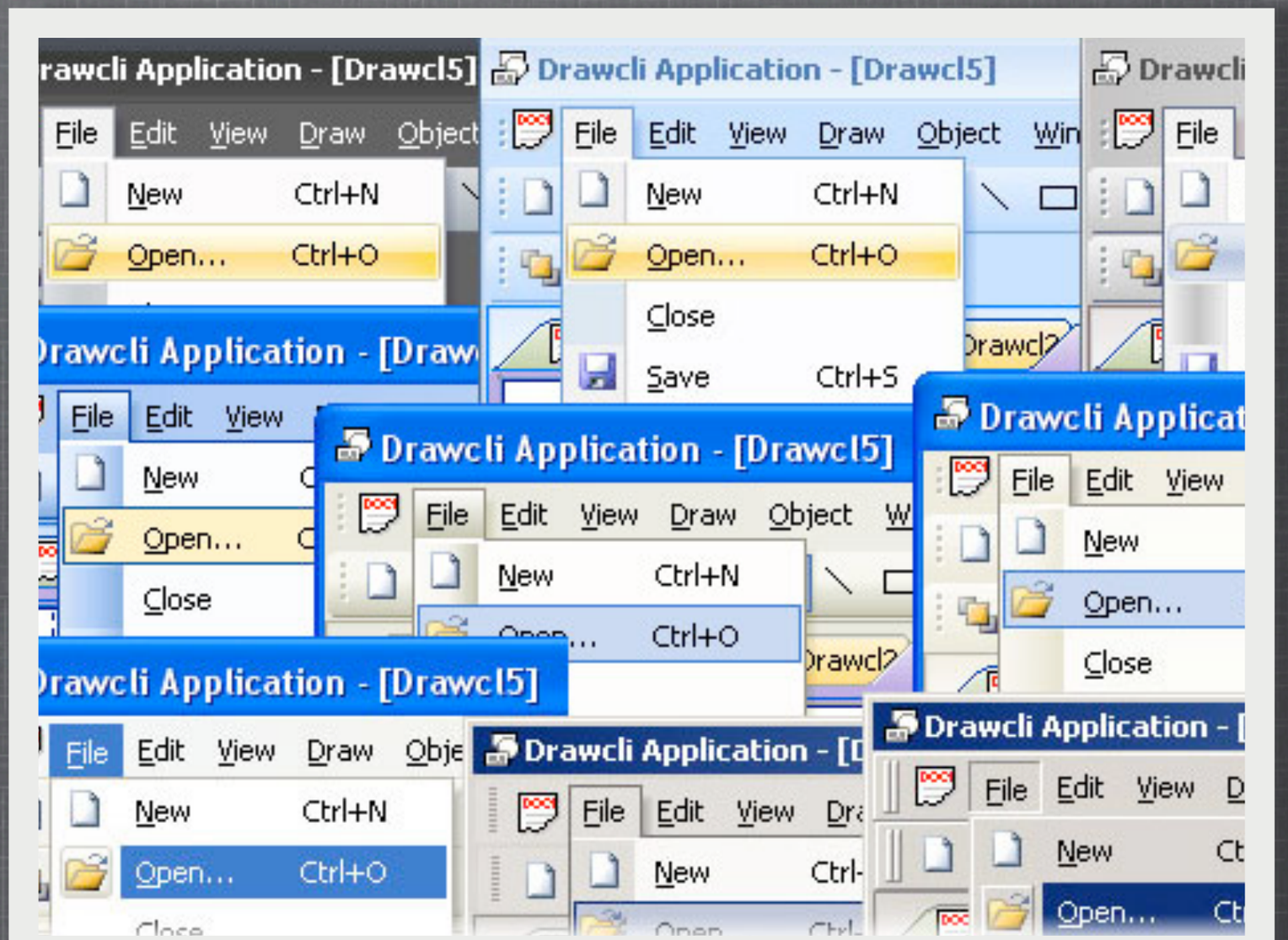
www.abiusx.com

دانشگاه شهید بهشتی
دانشکده مهندسی برق و کامپیوتر
بسیج دانشجویی
فروردین ۸۹

سلسله کارگاه های برنامه سازی تخصصی و کاربردی / کارگاه اول

GUI (GRAPHICAL USER INTERFACE) PROGRAMMING

برنامه نویسی واسط کاربری گرافیکی



عباس نادری (AbiusX.com)
دانشگاه شهید بهشتی
دانشکده مهندسی برق و کامپیوتر

فروردین ۸۹

تعریف اصطلاحات

خط فرمان / واسط کاربری

برنامه های گرافیکی که در محیط های پنجره ای اجرا می شوند، تفاوت های اساسی در همه ابعاد با برنامه های ساده خط فرمان (CLI) ـ که در دوره های رسمی برنامه نویسی، ساختن آنان را فرا می گیریم ـ دارند.
تقریبا تمام نرم افزارهای کاربردی و تجاری در قالب پنجره ای و گرافیکی هستند و برای تسلط به برنامه سازی کاربردی و تجاری، باید مسائل بسیاری که مربوط به برنامه سازی واسط کاربری گرافیکی هستند پوشش داده شوند.
علاوه بر آن، چالش اصلی دنیای نرم افزار امروز، یعنی چند سکویی (Cross Platform) بودن برنامه ها، مشکلات این عرصه را دوچندان کرده است، زیرا پنجره ها و امکانات گرافیکی تقریبا در هر سیستم عاملی نه تنها منحصر بفرد هستند، بلکه طراحی و قالبی کاملا متفاوت با سیستم عامل های دیگر دارند، بنابراین موضوع دیگری که باید پوشش داده شود، طراحی تطبیقی واسط کاربری برای تمامی سیستم عامل هاست.

سیستم های پنجره ای

سیستم پنجره ای (Window System) یکی از بخش های هر سیستم عامل است که بر روی واسط برنامه سازی (API)سیستم قرار می گیرد و امکان اجرای برنامه های گرافیکی و پنجره ای را به سیستم عامل می دهد. سیستم پنجره ای در سیستم عامل های مختلف به شرح زیر است:

ویندوز اکس پی به قبل : درون کار

ویندوز ویستا و 7: Desktop Window Manager (DWM) + Aero

مک ایکس : Aqua + Cocoa or Carbon

لینوکس : X Window System + KDE or GNOME or etc

بقیه : انحصاری یا موارد بالا

در طراحی های دقیق، یک سیستم پنجره ای خود از یک یا چند API به همراه یک یا چند دسکتاپ تشکیل شده است.

ثبت اجزا

در یک سیستم پنجره ای، هر برنامه ای از اجزای (Component) گرافیکی و غیرگرافیکی متعددی تشکیل شده است. برنامه باید برای داشتن هرکدام از این اجزا، آنرا به همراه مشخصات کاملش در سیستم عامل (سیستم پنجره ای) ثبت (Register) نماید. سپس سیستم عامل منابع مورد نیاز را در اختیار آن جزء از برنامه خواهد گذارد.

به هر جزء یک شماره دستگیره (Handle) اختصاص داده می شود. سیستم عامل جزء را با شماره آن می شناسد و برنامه های دیگر نیز می توانند به واسطه این شماره، جزئی از یک برنامه دیگر را تغییر دهند (برنامه هایی که رمز زیر ستاره ها را می خوانند از این دسته هستند) نرم افزارهای مدیریت وظیفه هر سیستم عاملی قادر به نمایش دادن تعداد اجزا و مولفه های گرافیکی هر برنامه هستند.

این اجزا، در محیط های مختلف با نامهای مختلف نامیده می شوند از جمله نام های : Control, Widget, ActiveX, OCX, Component,...

نخ بندی

در سیستم عامل های چندوظیفه ای (Multitasking) معمولا هر پروسه (Process) که یک اجرا از یک برنامه است، خود از یک تا چند نخ (Thread) تشکیل شده است که از منظر تئوری به صورت موازی اجرا می شوند. نخ بندی در برنامه های واسط گرافیکی تاحدودی الزامیست، زیرا هنگامی که برنامه مشغول رسم ظاهر خویش باشد، قطعا نمی تواند همزمان ورودی کاربر را نیز دریافت و پردازش و اعمال کند و همزمان فرآیند خود را نیز انجام دهد.

نخ بندی نیز در انحصار سیستم عامل است، یعنی هر برنامه ای که یک نخ جدید می خواهد، از سیستم عامل درخواست یک نخ می کند و تابعی از کد خود را در آن نخ به اجرا می گذارد. در اکثر موارد هر پروسه چندین نخ را به صورت خودکار به کار می گیرد بدون اینکه کاربر آن و حتی برنامه نویس آن مطلع باشد.

هر برنامه در ابتدای اجرا، یک نخ اصلی (Main Thread) دارد که وظیفه راه اندازی پروسه را نیز بر عهده دارد.

حلقه رخداد

هر برنامه واسط کاربری (غیر CLI) هنگامی که اجرا می گردد، ابتدا مولفه های اساسی خود را در سیستم ثبت می کند، مقداردهی و کارهای اولیه را انجام می دهد (این دو مرحله معمولا به صورت خودکار توسط چهارچوب برنامه سازی انجام می گیرد و برنامه نویس از آنان اطلاعی ندارد) و سپس وارد حلقه رخداد (Event Loop) یا حلقه اصلی (Main Loop) می شود.
برنامه در حلقه رخداد می ماند تا هنگامی که رخداد پایان را از سیستم عامل دریافت نماید یا خود آنرا تولید کند، سپس کارهای پایانی را انجام داده، بسته می شود.
تنها کار حلقه رخداد، پیام رسانیست. این حلقه مدام پیامهایی را از سیستم عامل می گیرد و به مولفه های مختلف برنامه می فرستد، از این رو نام های Message Dispatcher, Message Loop, Message Pump نیز برای این حلقه به کار می روند.
حلقه رخداد در تمامی چهارچوب ها به صورت ضمنی نوشته شده و وجود دارد.

چندوظیفگی پاره پاره

اکثر سیستم عامل های رایج امروزی، چندوظیفگی پاره پاره (Preemptive Multitasking) انجام می دهند. در این نوع از چند وظیفگی، سیستم عامل در هر لحظه دلخواه، اجرای یک برنامه را قطع می کند و پروسه دیگری را به اجرا در می آورد. معمولا این انقطاع در داخل حلقه رخداد برنامه اتفاق میافتد و مشکلی نیز برای برنامه ایجاد نمی کند، زیرا در این مدل برنامه مانند دستیاریست که کناری نشسته تا صدایش بزنند و پیامی به وی دهند. او سپس پیام را به بخش مربوطه منتقل می کند و کار آنرا انجام می دهد، آنگاه دوباره در جای خود می نشیند.

سیستم عامل به صورت خودکار حلقه رخدادهای برنامه ها را کنترل می کند، یعنی در موقع لزوم به آنها پیام استراحت می فرستد و کار آنها را قطع می کند و در موقع لازم به آنها پیام بیدارباش فرستاده، کار آنها را مجددا شروع می کند.

برنامه نویسی رخدادگرا

از آنجایی که تا پیامی به پروسه نرسد، برنامه مفیدی نیز اجرا نمی شود، برنامه نویسی واسط گرافیکی به برنامه نویسی رخدادگرا (Event-Oriented) نیز شهرت دارد.
هنگامی که یک پیام به حلقه رخداد می رسد، حلقه آنرا تفسیر می کند و به مولفه مربوطه ارسال می کند. در این هنگام برای مولفه رخدادی پیش آمده است و می تواند قطعه کد خاصی را برای مدیریت این رخداد (Event Handling) اجرا کند. بنابراین در برنامه نویسی واسط کاربری، تعدادی تابع برای مدیریت تعدادی رخداد تعریف می شوند و در زمان مناسب در نخ مناسب خود، اجرا خواهند شد.
درک این نکته که اینگونه برنامه، تفاوتی ماهوی با برنامه های خط فرمان دارد، بسیار حیاتیست زیرا در اینگونه برنامه ها، برنامه از خطی خاص از کد شروع به اجرا نمی شود و در خطی خاص نیز به پایان برسد، بلکه هر قطعه آن در زمانی خاص بنا بر شرایطی خاص ممکن است اجرا شود یا نشود، بنابراین توسعه و خطایابی آن دگرگون خواهد بود.

مدیریت رخداد

مدیریت رخداد در هر مولفه از سیستم واسط کاربری گرافیکی (GUI Component) توسط یک تابع به صورت ضمنی توسط چهارچوب برنامه سازی یا غیرضمنی توسط خود برنامه نویس (یا ترکیبی از این دو حالت) انجام می گیرد.
این تابع رخداد را از حلقه رخداد گردان دریافت می کند، نوع آنرا تشخیص می دهد، پارامترهای آنرا بدست می آورد و اگر برنامه ساز کد خاصی برای آن رخداد در نظر گرفته بود، آنرا اجرا می سازد.
به عنوان مثال هنگامی که رخداد کلیک ماوس اتفاق میافتد، این تابع رخداد را تشخیص می دهد، پارامتر های آن (مختصات و دکمه ها) را بدست می آورد و به توابعی که برنامه نویس مشخص کرده در صورت کلیک شدن بر روی این مولفه خاص (مثلا دکمه ای بر روی صفحه) اجرا شوند، ارسال می کند. همین تابع در صورت دریافت رخداد کیبورد، کاری مشابه انجام می دهد ولی در نهایت کدی که برنامه نویس برای حالت کیبورد نوشته است، اجرا می کند.

انواع رخداد

انواع معمول رخداد عبارتند از:
۱, رخداد رسم (Paint Event) : اینگونه رخدادها باعث می شوند هر برنامه یک ظاهر گرافیکی در سیستم داشته باشد.
۲, رخداد ورودی (ماوس و کیبورد) : اینگونه رخدادها تعامل کاربر با برنامه را تنظیم می کنند.
۳, رخداد تغییرات (Modification Event) : اینگونه رخدادها تغییر چیزی از برنامه، مانند تغییر ابعاد یک پنجره یا تغییر محتوی یک جعبه متن (Textbox) را مشخص می سازند.
۴, رخدادهای سیگنالی (Signals) : اینگونه رخدادها معمولا بوجود آمدن، بسته شدن و اینگونه موارد مولفه ها را کنترل می کنند.

سلسله مراتب رخدادها

اکثر رخدادهایی که بر روی یک مولفه اجرا می شوند، سلسله مراتبی مدیریت می شوند. به عنوان مثال دکمه ای را بر روی یک لیست در یک پنجره در نظر بگیرید که کاربری بر روی آن کلیک می کند. این رخداد کلیک توسط سیستم عامل، حلقه رخداد و تابع مدیریت رخداد به ترتیب به دکمه، به لیست و به پنجره در مختصات های متفاوت (نسبت به مکان هرکدام) ارسال می گردد.

اگر هرکدام از این مولفه ها، هنگام پردازش این رخداد، نتیجه مثبت برگرداند، بدین معنیست که رخداد را کاملا پردازش و مصرف کرده و نباید به مولفه بعدی ارسال شود. اما اگر نتیجه منفی بازگرداند، یعنی این رخداد متعلق به من نبود (انحصارا) و باید به مولفه در اولویت بعدی نیز ارسال گردد.

کلیت این فرآیند نیز معمولا به صورت ضمنی و با تغییر در مشخصات (Properties) یک مولفه کنترل می شود و احتیاج به برنامه نویسی ندارد.

رخداد رسم

رخداد رسم، چیزیست که باعث می شود یک مولفه در سیستم ظاهر گرافیکی داشته باشد. سیستم در بازه های زمانی معینی، برای هرکدام از مولفه ها که در تصویر قرار دارند، رخداد رسم ارسال می کند (پنجره ای که در پشت قرار داشته باشد، در سیستم های مدرن رخداد رسم دریافت نمی کند)
هر مولفه ای که رخداد رسم را دریافت کرد، با استفاده از ابزار رسم سیستم (که در هر سیستم پنجره ای متفاوت است) شروع به رسم شکل خود بر روی صفحه می کند.
به عنوان مثال یک دکمه شروع به رسم یک مستطیل گرد شده می کند و داخل آن متن خود و افکت های سایه و غیره خود را نیز اعمال می کند.
در همه چهارچوب های (Frameworks) برنامه سازی، تعدادی مولفه از پیش آماده شده وجود دارد که علاوه بر کارهای بسیار دیگر، ظاهر خود را نیز به صورت ضمنی رسم می کنند و اصلا احتیاجی نیست برنامه نویس کد مربوط به رخداد رسم آنها را پیاده کند، ولی برای تعریف یک مولفه (Widget) جدید، باید کد را نوشت.

مشخصات

در اکثر چهارچوب های برنامه سازی واسط گرافیکی که به صورت شی ء گرا نوشته شده اند، هر مولفه یک شی ء گسترش یافته (شیئی با امکاناتی بیشتر از یک شی ء) است. بسته به چهارچوب مورد نظر، مولفه ممکن است مشخصات (Properties)، سیگنال و شیار (Signal/Slot) و امکانات دیگری داشته باشد.

مشخصات (Properties) در بسیاری از چهارچوب ها پیاده شده اند و ویژگی های یک مولفه را در بر می گیرند. به عنوان مثال یک دکمه بر روی یک صفحه معمولا حدود ۳۰ مشخصه دارد، از جمله محل قرار گیری و ابعاد آن، متن روی آن، رنگ آن، قفل بودن یا نبودن آن، فونت متن آن و ...

تفاوت اصلی مشخصات با متغیرهای عضو (Member Variables) یک مولفه در آنست که مشخصات در طراحی ظاهری (Visual Design) نیز قابل تغییر هستند و معمولا به داده مولفه کاری ندارند، بلکه ظاهری هستند.

طراحی بصری

تقریبا به همراه هر چهارچوب برنامه سازی واسط گرافیکی، یک یا چند برنامه طراحی بصری وجود دارد که با استفاده از آن می توان تعدادی مولفه را در یک پنجره چید و مشخصات آنرا به طور دلخواه تغییر داد، سپس آنرا در قالبی خاص ذخیره و به کد تبدیل کرد.

اینگونه طراحی، صرفه جویی قابل ملاحظه ای در زمان برنامه ساز می کند، زیرا طراحی واسط گرافیکی کاریست بصری و هنگامی که به کد در می آید حجم انبوهیست از کدهای نسبتا تکراری.

اگر اینگونه برنامه ها و محیط های طراحی بصری درون کار محیط برنامه سازی (IDE) باشند، معمولا به شما امکان ایجاد تابع هایی از کد برای مدیریت یک رخداد خاص از یک مولفه را نیز می دهند، به این صورت که با کلیک بر روی مولفه، لیست رخدادهای آن نمایان می شود و هنگامی که بر روی یکی از آنها کلیک کنید، خودکار تابعی نوشته می شود و در مدیریت رخداد قرار می گیرد و شما به آن هدایت می شوید.

جهان چوبی ها

چهارچوب ها

چهارچوب های بسیاری برای تولید برنامه های واسط گرافیکی وجود دارند. هر چهارچوب مزایا و معایب خاص خود را دارد. برخی چهارچوب ها در حد مجموعه ای از ویجت ها هستند و برخی کلیه کتابخانه ها و کدهای لازم جهت تولید یک برنامه معمول را نیز در بر می گیرند. معروف ترین این چهارچوب ها به شرح زیرند:

دات نت (.NET Framework) که به همراه Microsoft Visual Studio به عنوان محیط طراحی اصلی استفاده می شود.

کیوت (Qt) که به همراه Qt Creator به عنوان محیط طراحی استفاده می شود.

جی تی کا (GTK) که بدون محیط طراحی استاندارد استفاده می شود و روالیست.

کربن (Carbon) که به همراه محیط طراحی Xcode استفاده می شود و روالیست.

کاکائو (Cocoa) که به همراه محیط طراحی Xcode استفاده می شود.

ای دبلیو تی (AWT) که به عنوان Abstract Window Toolkit در جاوا ا ست.

سوییینگ (Swing) که به عنوان مجموعه جدیدتری برای جاوا شناخته می شود.

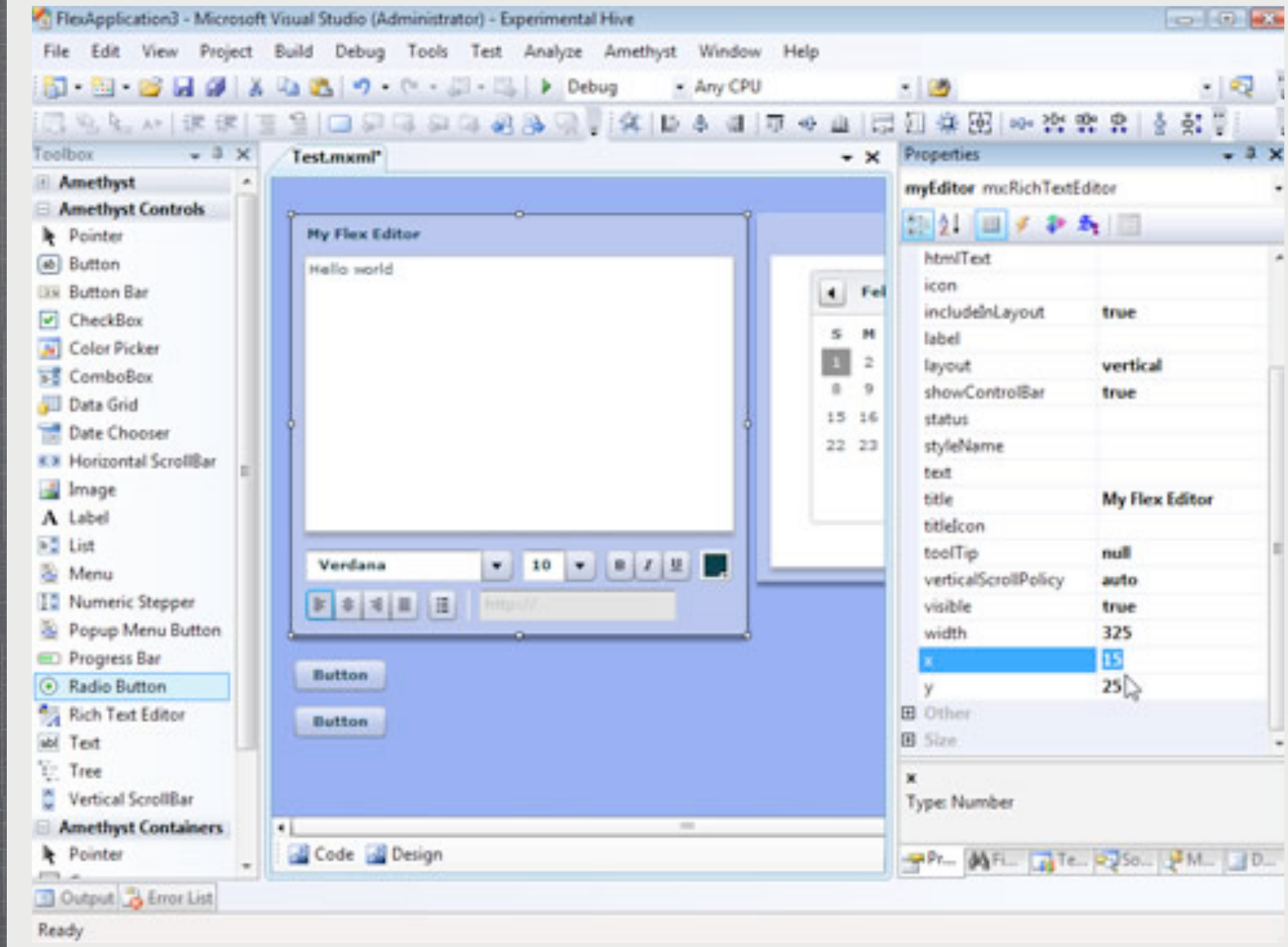
اس دبلیو تی (SWT)که به عنوان استاندارد Eclipse برای جاوا استفاده می شود.

.NET Framework

چهارچوب دات نت، از حوالی سال ۲۰۰۰ توسط مایکروسافت به سکوی اصلی توسعه نرم افزار برای ویندوز تبدیل شد. در حال حاضر عمده برنامه های این چهارچوب تنها بر روی ویندوز اجرا می شوند و اگر برنامه ای با نسخه جدیدی این چهارچوب (مثلا ۳,۵) نوشته شود، باید کل چهارچوب در مقصد نصب گردد تا برنامه اجرا شود (۳۵۰ مگابایت)

این چهارچوب تقریبا از نظر امکانات تکمیل است و توسعه نرم افزار با محیط Visual Studio که محیط اصلی توسعه با این چهارچوب است، بسیار راحت است. از مهمترین مضرات این چهارچوب آنست که توسعه دهندگان به دلیل سادگی بیش از حد، بدون درک روش توسعه خود نکات حیاتی را از قلم می اندازند.

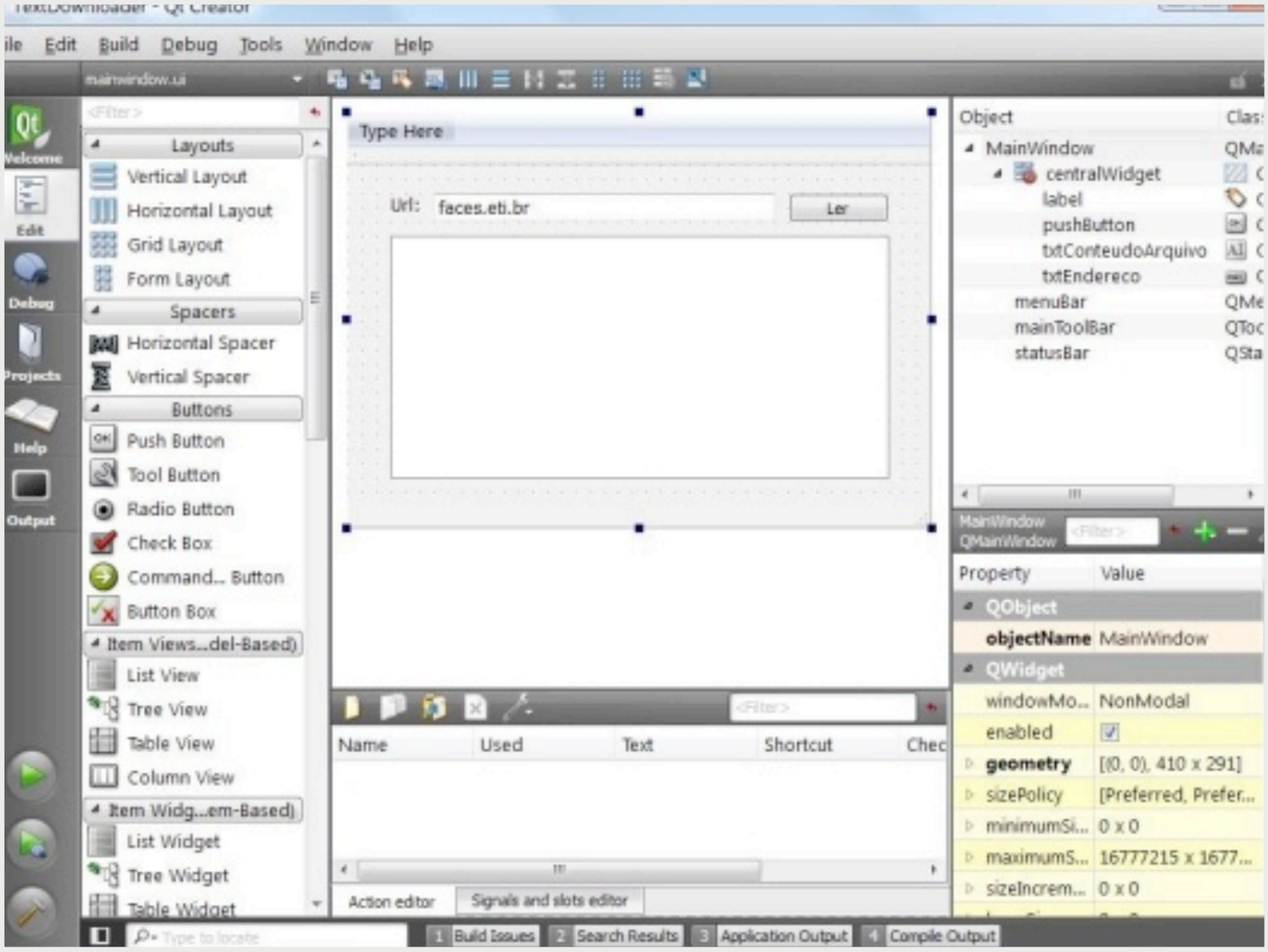
محیط Visual Studio شامل زبان های VB.NET, VC#.NET, VC.NET است که ویژوال سی پلاس پلاس دات نت آن، به غیر از چهارچوب دات نت از چهارچوب های MFC و هسته ویندوز نیز پشتیبانی می کند.



سه‌شنبه 20 تیر 2010

کیوت Qt

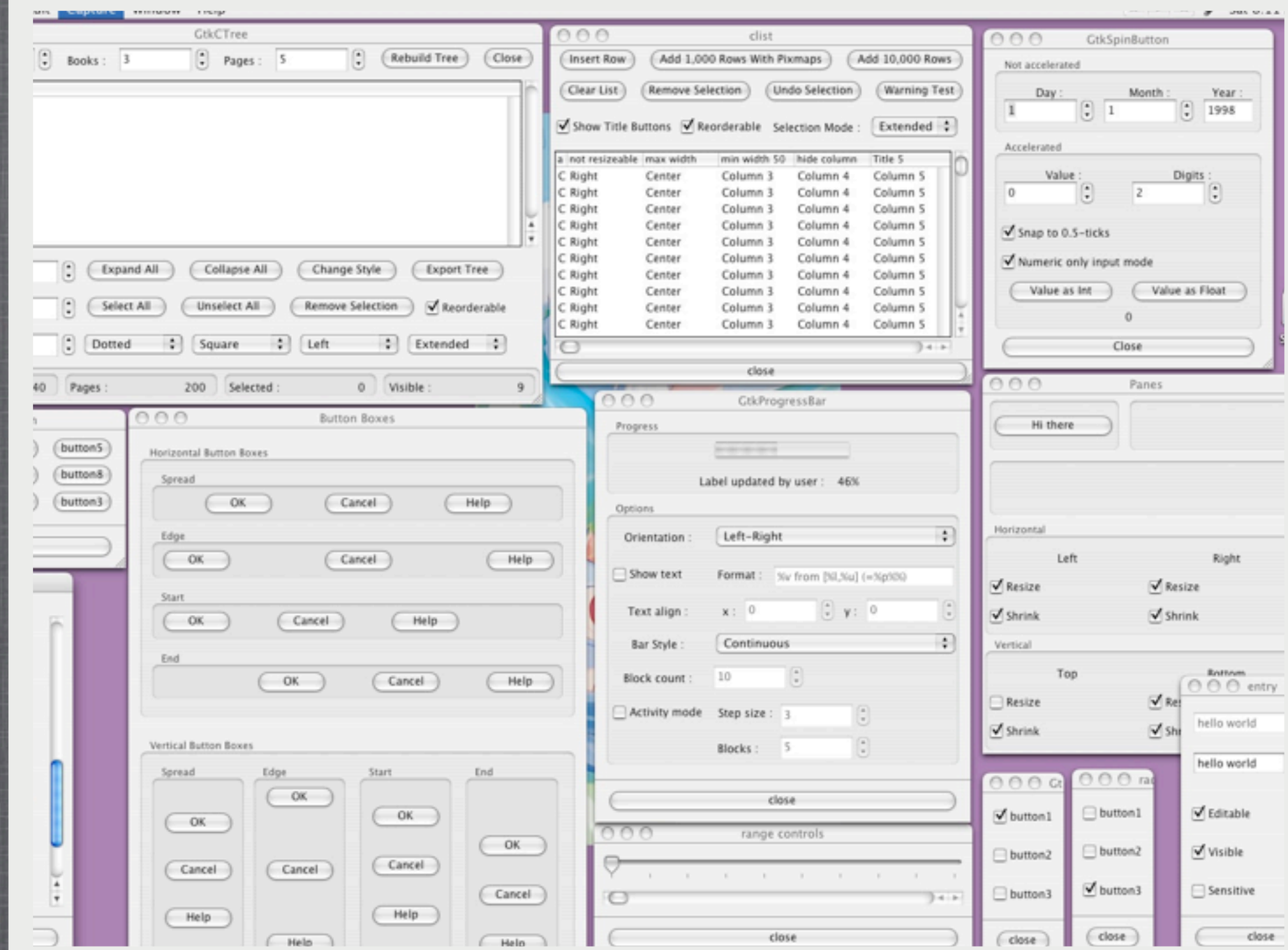
کیوت چهارچوبیست که TrollTech تهیه کرده و نوکیا حق آنرا خریده است. کل میزکار KDE که یکی از دو میزکار (و میزکار حرفه ای) لینوکس است، با استفاده از Qt برنامه سازی شده است. کیوت چندسکوییست و حتی بر روی انواع موبایل نیز کار می کند.
هم قابلیت پیوند ایستا دارد و هم پویا، و از نظر امکانات در حال حاضر از جاوا نیز قدرتمندتر گشته است.
کیوت تقریبا برای همه زبانهای برنامه نویسی تبدیل (Bind) شده است ولی زبان اصلی آن C++ است.به همراه Qt SDK نرم افزارهای بسیاری برای ایجاد برنامه توسط کیوت وجود دارد از جمله IDE کامل Qt Creator. در حال حاضر کیوت از منظر کارایی و سرعت از مابقی چهارچوب ها با اختلاف جلو است. سیستم راهنما و آموزش کامل نیز به همراه آن وجود دارد. مشکل اصلی کیوت آنست که نسخه C++ آن کمی دشوار است (کلا زبان C++ دشوار است)


File Edit Build Debug Tools Window Help
mainwindow.ui
Welcome
Edit
Debug
Projects
Help
Output
Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Form Layout
Spacers
Horizontal Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Command... Button
Button Box
Item Views...del-Based)
List View
Tree View
Table View
Column View
Item Widg...em-Based)
List Widget
Tree Widget
Table Widget
Type Here
Url:
faces.eti.br
Ler
Name
Used
Text
Shortcut
Action editor
Signals and slots editor
Object
MainWindow
centralWidget
label
pushButton
txtConteudoArquivo
txtEndereco
menuBar
mainToolBar
statusBar
MainWindow
QMainWindow
Property
Value
QObject
objectName
MainWindow
QWidget
windowMo...
NonModal
enabled
geometry
[(0, 0), 410 x 291]
sizePolicy
[Preferred, Prefer...
minimumSi...
0 x 0
maximumS...
16777215 x 1677...
sizeIncrem...
0 x 0
Type to locate
1 Build Issues
2 Search Results
3 Application Output
4 Compile Output

جی تی کی GTK

جی تی کی مخفف Gimp Toolkit است، ابتدا چهارچوبی بوده که نرم افزار معروف Gimp را با استفاده از آن ساخته اند و سپس گسترش یافته به یک چهارچوب اساسی تبدیل شده که میزکار محبوب لینوکس، GNOME کلا با استفاده از آن طراحی شده است. جی تی کی روالیست و امکانات کاملی ندارد، همچنین محیط برنامه سازی نیز به همراه آن (رسمی) وجود ندارد و به مانند اکثر چهارچوبهای متن باز، تکه تکه باید جمع و سرهم شود.

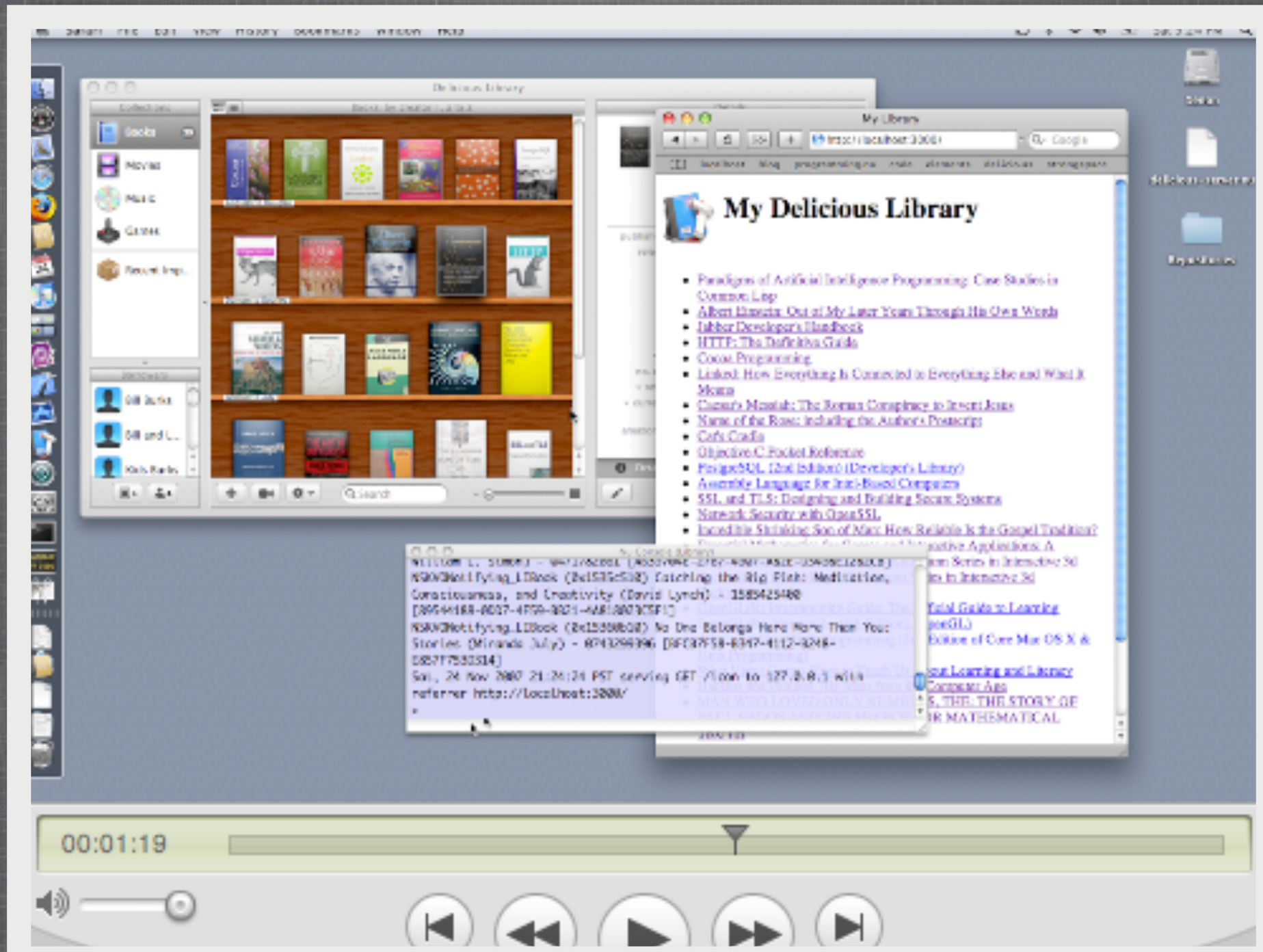
به مانند کیوت، جی تی کی نیز برای تمام زبانهای برنامه سازی تبدیل (Bind) شده است ولی زبان اصلی آن C است. جی تی کی به دلیل سادگی از کیوت محبوبیت بیشتری دارد ولی امکانات آن به شدت کمتر و نامنسجم تر است. همچنین جی تی کی چند سکویی برای ویندوز، لینوکس و مک است.


GtkCTree
Books : 3
Pages : 5
Rebuild Tree
Close
Expand All
Collapse All
Change Style
Export Tree
Select All
Unselect All
Remove Selection
Reorderable
Dotted
Square
Left
Extended
Pages : 200
Selected : 0
Visible : 9
clist
Insert Row
Add 1,000 Rows With Pixmaps
Add 10,000 Rows
Clear List
Remove Selection
Undo Selection
Warning Test
Show Title Buttons
Reorderable
Selection Mode : Extended
a not resizeable max width min width 50 hide column Title 5
C Right Center Column 3 Column 4 Column 5
close
GtkSpinButton
Not accelerated
Day : Month : Year :
1 1 1998
Accelerated
Value : Digits :
0 2
Snap to 0.5-ticks
Numeric only input mode
Value as Int
Value as Float
0
Close
Button Boxes
Horizontal Button Boxes
Spread
Edge
Start
End
OK
Cancel
Help
Vertical Button Boxes
GtkProgressBar
Progress
Label updated by user : 46%
Options
Orientation : Left-Right
Show text
Format : %v from [%l,%u] (=%p%%)
Text align : x : 0 y : 0
Bar Style : Continuous
Block count : 10
Activity mode
Step size : 3
Blocks : 5
close
range controls
Panes
Hi there
Horizontal
Left
Right
Resize
Shrink
Vertical
Top
entry
hello world
Editable
Visible
Sensitive
button1
button2
button3
button5
button8

کربن و کاکائو

دو چهارچوب معروف مک که تنها بر روی مک قابل اجرا هستند و زبان آنها Objective-C است.کربن و کاکائو توسط محیط برنامه سازی Xcode قابل استفاده هستند و از انعطاف و قدرت بالایی برخوردار هستند. همچنین این دو چهارچوب امکانات گرافیکی و نمایشی خارق العاده مک را کاملا در اختیار برنامه ساز قرار می دهند.

کربن چهارچوب قدیمی تریست و روالیست و کاکائو جدیدتر و شی گراست. امروزه کاکائو نیز تقریبا از تمامی امکانات کربن پشتیبانی می نماید. از آنجایی که سیستم عامل مک امکاناتی چندین برابر دیگر رقیبان خود دارد، برنامه سازی با استفاده از این دو چهارچوب برنامه های بسیار مطلوبتری از نمونه های تولید شده توسط چهارچوب های چند سکویی (مانند کیوت) برای مک تولید می کند.


My Delicious Library
My Libraries
My Delicious Library
00:01:19

AWT, Swing, SWT

دو چهارچوب گرافیکی جاوا که یکی از ابتدا به همراه جاوا بوده و دیگری با زیباتر شدن ظاهر برنامه ها تولید گشته است. هر دو به عنوان استاندارد همراه جاوا هستند و چند سکویی اجرا می شوند. امکانات گرافیکی هردو نسبتا خوب است و ابزار جانبی برای طراحی بصری توسط هردو وجود دارد.

نرم افزارهای جاوا به دلیل مفسری بودن به راحتی در اکثر سیستم ها با سرعت قابل قبولی اجرا می شوند، ولی با به عرصه آمدن چهارچوب های متن باز (مانند کیوت و جی تی کی) و قدرت و سرعت بسیار بالاتر آنان، نرم افزارهای جاوایی مخصوصا در ابعاد بزرگ به سرعت در حال منزوی شدن هستند.

چهارچوب SWT توسط تیم Eclipse (محبوب ترین محیط برنامه سازی متن باز) برای جاوا طراحی شده و خود اکلیپس محیط طراحی بصری برای نرم افزارهای گرافیکی جاوا با این چهارچوب را نیز در بر می گیرد. این چهارچوب غنی تر و از لحاظ ظاهری زیباتر از دوچهارچوب قبلیست.

WidgetsDemo

File Options Help

name

Spade, Sam
Gaukley, Gundel
Johnson, Don
Dubarray, Marie J
敬群, 黄

last 敬群
firs 黃
cate other
sex: Mal Fema

Ad Mo Re Cld

comm

這是GNU Classpath的
AWT實做在Kaffe VM運作的畫

敬群, 黃

Torrent Allocating...
Torrent Downloading...
Tracker Scrape Result : Scra

d Distributor

/download.torrent

Browse...

p Reset / Clear

Properties for XSWT

pe filter text

Info
Builders
Java Build Path
Java Code Style
Java Compiler
Javadoc Location
Project Reference

Java Build Path

Source Projects Libraries Order and Export

JARs and class folders on the build path:

swt.jar - /Users/daniel/D
swt-pi.jar - /Users/danie
JRE System Library [JVM 1.

Add JARs...
Add External JARs...
Add Variable...
Add Library...

Default output folder:

XSWT Browse...

Cancel OK

رسم سطح پایین

رسم در صفحه

در تمام محیط های پنجره ای، واسطی برای کار با کارت گرافیکی وجود دارد. به عنوان مثال در ویندوز و دات نت، واسط GDI توانایی عملیات رسم دوبعدی (و سه بعدی ساده) را داراست.
اینگونه واسط ها برای رسم ابتدایی یک ویجت استفاده می شوند و هنگامی که توسعه دهنده قصد طراحی یک ویجت جدید را داشته باشد، باید با استفاده از این واسط ها ویجت خود را رسم نماید (همانطور که ذکر شد).
واسط های هر سیستمی مزایا و معایبی دارند، به عنوان مثال GDI در ویندوز ویستا به بعد زیر نظر WDM کار می کند که این باعث کندی بیش از حد آن شده است.
چهارچوب کیوت این امکان را فراهم می سازد که با استفاده از OpenGL رسم ابتدایی ویجت ها را انجام دهید که بسیار سریعتر است.
به طور کلی رسم در صفحه از ویژگیهای سطح پایین سیستم های پنجره ایست که همواره برای نرم افزارهای چند سکویی ایجاد مشکل می نماید.

سلسله کارگاه های برنامه سازی تخصصی و کاربردی / کارگاه اول

GUI (GRAPHICAL USER INTERFACE) PROGRAMMING

برنامه نویسی واسط کاربری گرافیکی

سوالات، پیشنهادات؟

عباس نادری
me@AbiusX.com
دانشگاه شهید بهشتی
دانشکده مهندسی برق و کامپیوتر
فروردین ۸۹