

ITdevCon

European Delphi Conference

14, 15 november 2013 - VERONA (Italy)

Delphi Toolbox

La cassetta degli attrezzi ideale
per le applicazioni moderne

Marco Breveglieri

Software and Web Developer

Obiettivi

- Individuare le problematiche ricorrenti nello sviluppo delle applicazioni moderne
- Vedere esempi pratici di soluzioni, basate sulle risorse già disponibili in Delphi e/o di terze parti
- Metodologie efficaci per la strutturazione dei progetti
- Link per reperire tool, suggerimenti e/o altro
- Altre problematiche e tool di vostra conoscenza?
- Q&A
- Conclusioni

ITDevCon

European Delphi Conference

Contro il logorio della vita moderna

dello sviluppatore, s'intende...



Le problematiche ricorrenti

Le esigenze e le problematiche ricorrenti del team di sviluppo:

- Essere «produttivi» nella scrittura del codice
- Gestione delle impostazioni di configurazione
- Gestione delle «dipendenze» tra i componenti
- Esecuzione di Unit/Integration Test
- Documentazione del codice (per il team di sviluppo)
- Distribuzione finale del prodotto

Quali soluzioni

Le soluzioni possono essere ricercate

- Nella libreria RTL di Delphi, nella VCL e in FireMonkey
- Negli add on forniti in bundle con il prodotto
- Nelle librerie e componenti di terze parti (gratuite e a pagamento)

grazie a

- Ricco set di classi che fanno parte di RTL/VCL/FMX
- Rinvigorito mercato di componenti, tool e librerie
- Possibilità estremamente elevate di poter migrare codice sorgente anche molto datato

ITDevCon

European Delphi Conference

Alcune soluzioni...



Strumenti di produttività

Un paio di suggerimenti...

Add-on per l'IDE:

- **CnPack CnWizards**
(<http://www.cnpack.org/>)
- **TwoDesk Castalia**
(<http://www.twodesk.com/castalia>)
- Tool di refactoring built-in

Documentazione del codice:

- **DevJet Documentation Insight**
(<http://www.devjetsoftware.com/products/documentation-insight/>)

Gestione delle impostazioni di configurazione

Possiamo utilizzare le funzionalità già presenti in Delphi:

- **TPersistent/TComponent** per la definizione delle impostazioni da salvare
- Funzioni della **RTL** per il caricamento/salvataggio di istanze dei componenti
- Classi **Stream** per il trasferimento dei dati a/da disco, memoria, rete...

Vediamo un esempio...

Gestione delle dipendenze

Introduciamo la «Dependency Injection»:

- **Spring4D** (<http://www.spring4d.org>) fornisce (tra le altre cose) un «container» per Dependency Injection

Vediamo un esempio...

Test della logica delle applicazioni

E' importante scrivere ed eseguire Unit Test e Integration Test per verificare il proprio codice.

Possiamo utilizzare:

- **DUnit** (incorporato in Delphi) per scrivere le classi di test
- **DelphiMocks** (<https://github.com/VSoftTechnologies/Delphi-Mocks>) per velocizzare la creazione dei test

Vediamo un esempio...

Altre problematiche...

General Purpose (container, liste, generics, strutture dati, ...).

- **Spring4D**
- **DeHL** (<https://code.google.com/p/delphilhlplib/>) *discontinued

Data access

- **dORM** (<https://code.google.com/p/delphi-orm/>)

Profilazione e debugging

- **SamplingProfiler** (<http://www.delphitools.info/samplingprofiler/>)
- **FastMM** (<http://sourceforge.net/projects/fastmm/>)

Distribuzione

- **InnoSetup** (<http://www.jrsoftware.org/>)

...altre ancora?

ITDevCon

European Delphi Conference

Conclusioni

...e grazie!

