

linkedin.com/in/bartsokol 

bart-sokol.info 

[@bartsokol](https://twitter.com/bartsokol) 

EVOLUTION OF DEVELOPER

Od wczoraj do jutra

linkedin.com/in/bartsokol 

bart-sokol.info 

[@bartsokol](https://twitter.com/bartsokol) 

O MNIE SŁÓW KILKA

Bartosz Sokół
Lead Developer .NET
GFT Polska

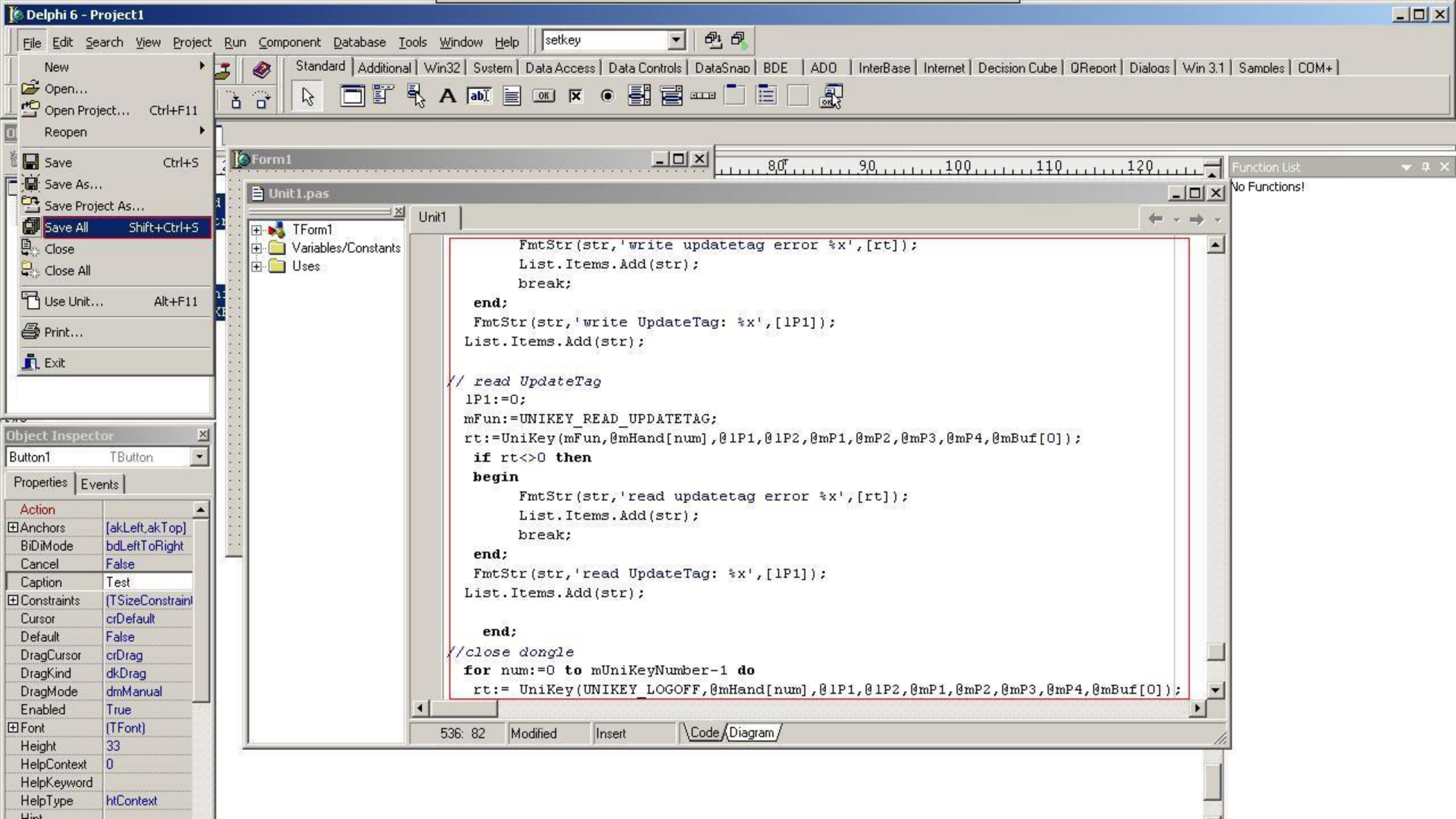
HISTORIA

Cofnijmy się o kilkanaście lat...

```
PROGRAM AUTOARRAY;  
USES CRT;  
VAR  
  A:ARRAY[1..10,1..10] OF INTEGER;  
  LINE:INTEGER;  
  COLUMN:INTEGER;  
  
BEGIN  
  {INPUT}  
  FOR LINE := 1 TO 10 DO  
    FOR COLUMN := 1 TO 10 DO  
      A[LIN, COLUMN] := RANDOM(90)+10;  
  
  {OUTPUT}  
  FOR LINE := 1 TO 10 DO  
    FOR COLUMN := 1 TO 10 DO
```

LATA 90'

- ❖ MS DOS, Windows 3.11, Windows 95
- ❖ Najpopularniejsze języki programowania: C, C++, (Turbo) Pascal
- ❖ Wiedza zdobywana z książek od nauczyciela informatyki (Marciniak!)
- ❖ Ukochane IDE – Turbo Pascal (jedyne co działa na moim 286)
- ❖ Najciekawszy projekt – własna nakładka graficzna na MS DOS. Na kartę graficzną Hercules. Z okienkami. I obsługą myszki. Szaleństwo.



File Edit Search View Project Run Component Database Tools Window Help

setkey

- New
- Open...
- Open Project... Ctrl+F11
- Reopen
- Save Ctrl+S
- Save As...
- Save Project As...
- Save All Shift+Ctrl+S
- Close
- Close All
- Use Unit... Alt+F11
- Print...
- Exit

Object Inspector

Button1 TButton

Properties Events

- | Action | |
|-------------|-----------------|
| Anchors | [akLeft,akTop] |
| BiDiMode | bdLeftToRight |
| Cancel | False |
| Caption | Test |
| Constraints | (TSizeConstrain |
| Cursor | crDefault |
| Default | False |
| DragCursor | crDrag |
| DragKind | dkDrag |
| DragMode | dmManual |
| Enabled | True |
| Font | (TFont) |
| Height | 33 |
| HelpContext | 0 |
| HelpKeyword | |
| HelpType | htContext |
| Hint | |

Form1

Unit1.pas

- TForm1
- Variables/Constants
- Uses

Unit1

```
        FmtStr(str,'write updatetag error %x',[rt]);
        List.Items.Add(str);
        break;

    end;
    FmtStr(str,'write UpdateTag: %x',[lP1]);
    List.Items.Add(str);

// read UpdateTag
lP1:=0;
mFun:=UNIKEY_READ_UPDATETAG;
rt:=UniKey(mFun,@mHand[num],@lP1,@lP2,@mP1,@mP2,@mP3,@mP4,@mBuf[0]);
    if rt<>0 then
    begin
        FmtStr(str,'read updatetag error %x',[rt]);
        List.Items.Add(str);
        break;

    end;
    FmtStr(str,'read UpdateTag: %x',[lP1]);
    List.Items.Add(str);

    end;
//close dongle
for num:=0 to mUniKeyNumber-1 do
    rt:= UniKey(UNIKEY_LOGOFF,@mHand[num],@lP1,@lP2,@mP1,@mP2,@mP3,@mP4,@mBuf[0]);
```

Function List

No Functions!

536: 82

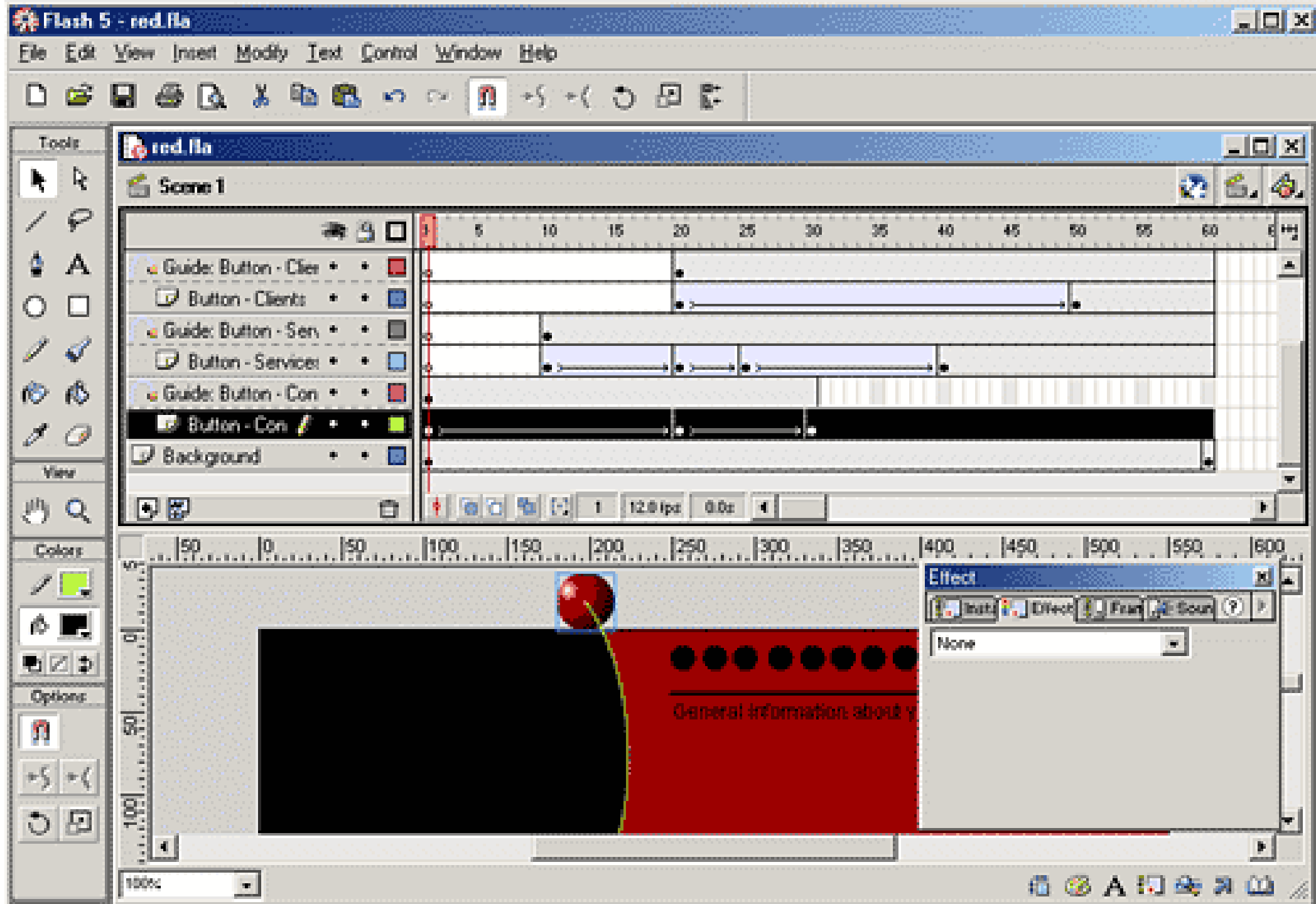
Modified

Insert

Code Diagram

ZBLIŻA SIĘ ROK 2000

- ❖ Delphi zastępuje Turbo Pascala
- ❖ Internet szturmem zdobywa świat
- ❖ Pojawiają się pierwsze kafejki internetowe
- ❖ Windows 98 króluje na komputerach domowych
- ❖ Masz „stałkę” – masz kumpli i wszyscy siedzą u Ciebie, nawet jak Cię nie ma w domu
- ❖ W międzyczasie instaluję Monkey Linux na moim 486SX z 2 MB RAM i 41 MB /5”/ dyskiem (przyniesiony z liceum na 12 dyskietkach). Web development, w Vi i Lynx. Cool!



POCZĄTKI XXI WIEKU

- ❖ `<marquee><blink>`Na horyzoncie pojawia się HTML 4.0 i CSS`</blink></marquee>`
- ❖ PHP w wersji 3 podbija Internet...
- ❖ ...podobnie jak Flash...
- ❖ ...i DHTML...
- ❖ ...i Google!
- ❖ Pora zacząć na tym zarabiać – sprzedaję pierwszą stronę w PHP. *(Do czego służy PHP na stronie? Prawidłowa odpowiedź zostanie nagrodzona gadżetem)*



Home



Removable
media



Buy new
software



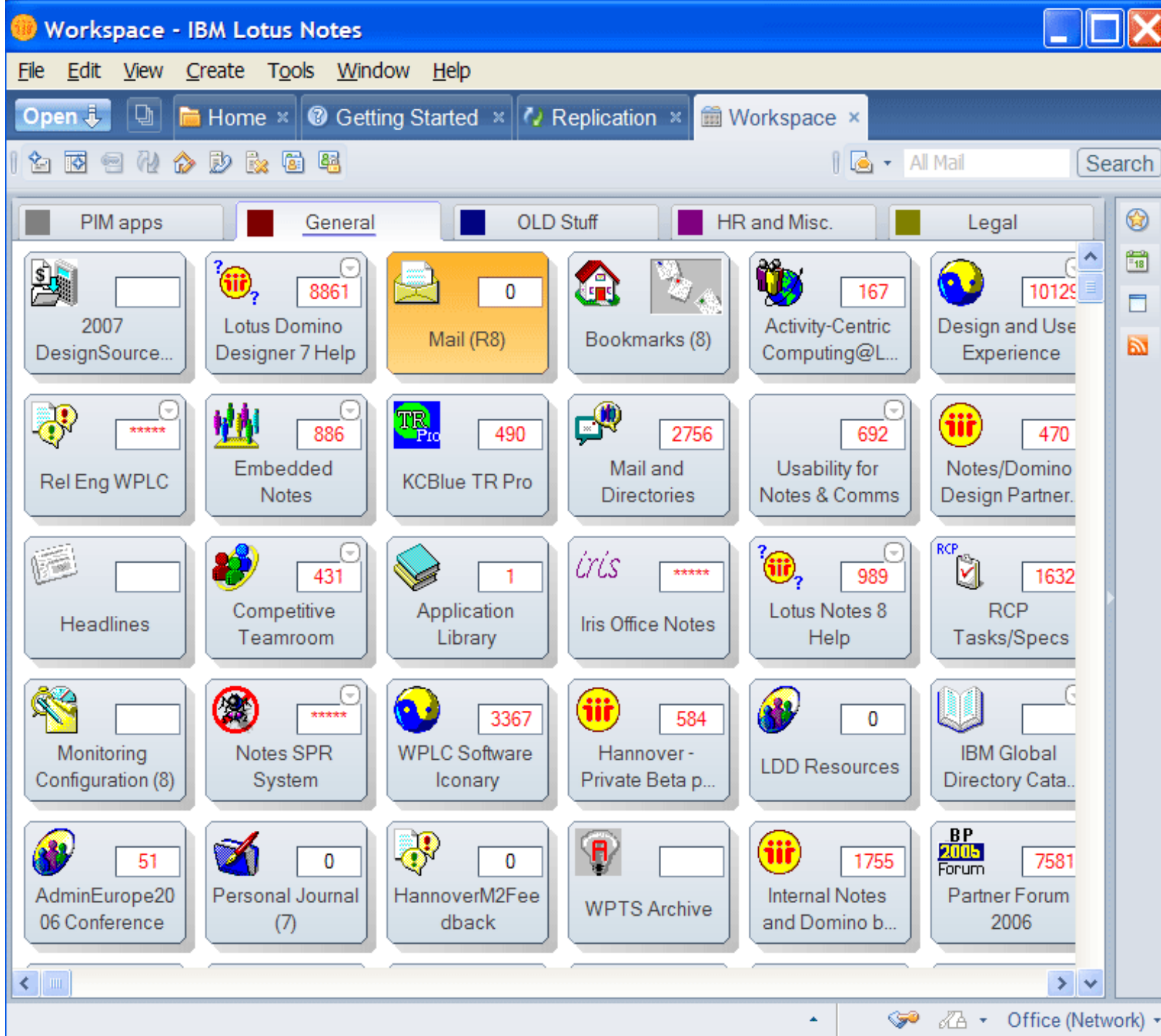
MandrakeClub

KDE 3

- Licq (KDE)
- ★ Mozilla
- World Clock
- The GIMP
- Licq
- Amusement
- Applications
- Configuration
- Documentation
- > What to do?
- Multimedia
- Networking
 - Chat
 - BlueFish
 - File transfer
 - Bookmarks Editor
 - Instant messaging
 - Galeon
 - IRC
 - Konqueror Web Browser
 - Mail
 - ★ Mozilla
 - News
 - ★ Mozilla Composer
 - Remote access
 - Quanta Plus
 - WWW
 - Screem
- Office
- Terminals
- Control Center
- Home (Personal Files)
- Bookmarks
- Recent Documents
- Quick Browser
- Run Command...
- Configure Panel
- Lock Screen
- Logout

KILKA LAT PÓŹNIEJ

- ❖ Powstaje projekt Firebird. Znany później jako Phoenix. *Kto wie jak się teraz nazywa?*
- ❖ Windows XP jest już standardem. Tak wiem, gdzieś do dziś.
- ❖ Pojawia się takie coś jak .NET
- ❖ Poznaję nowe języki. Ada 95, anyone?
- ❖ Zakochuję się w Pythonie. Bez skojarzeń proszę.
- ❖ Porzucam Windowsa na rzecz Linuksa. Co tydzień inna dystrybucja. Mandrake, Debian, Corel, Slackware, (...)



JA NIE CHCĘ BYĆ PROGRAMISTĄ. ALE MUSZĘ.

- ❖ Nie podobało mi się na studiach. Zwłaszcza programowanie. Porzucam studia żeby zostać fotografikiem.
- ❖ Kilka miesięcy później dostaję pracę jako programista.
- ❖ Dowiaduję się że niektórzy ciągle piszą w Turbo Pascalu. Zmieniam pracę.
- ❖ Poznaję kolejne języki. Formula+, anyone? Lotus Formula? VBA wiem, każdy...
- ❖ Tests? Where I'm working I don't need tests!
- ❖ W tzw. między czasie rzeźbię w PeHaPie, Pythonie, Bashu, Linuksie, JavaScriptcie, HTMLu, CSSie i czymkolwiek co się nawinie...



REWOLUCJA

- ❖ Syn marnotrawny wrócił na uczelnię. Poznał co to C#. I NET. I kilka innych języków i platform.
- ❖ Znudziło mi się klepanie ERP-ów. Rzucam robotę. Robimy startupa! Efekt?
- ❖ Eeee, to się nie uda. Było za wcześnie. Przenoszę się do innego miasta. Żeby pisać nowego ERPa.
- ❖ Nawracam się na .NETa
- ❖ Staję się ekspertem od WPF i MVVM. Rok temu jeszcze nie wiedziałem co to. Co się nie da, jak się da!
- ❖ Pojawiam się na moim pierwszym spotkaniu PG .NET! 😊
- ❖ W podziękę Microsoft zaczyna udostępniać kod .NET na licencji OSS. How cool is that?

TERAZ

Live on Twitter...

OMG CO SIĘ TU DZIEJE?!

- ❖ Komputer w kieszeni każdego z nas
- ❖ Internet w (prawie) każdym miejscu
- ❖ Setki tweetów na godzinę do przeczytania
- ❖ Każdy może programować
- ❖ Nowy framework w JS co 15 sekund
- ❖ Nieskończona ilość kodu we wszystkich językach dostępna na GitHubie
- ❖ Zapotrzebowanie na programistów znacząco przewyższa podaż
- ❖ Ciągle nie mam swojego hoverboarda 😞

TO CO SIĘ DZIEJE

- ❖ JavaScript najpopularniejszym językiem programowania
- ❖ Wszystko działa w chmurze, chyba że akurat chmura nie działa
- ❖ Języki funkcyjne szturmem zdobywają programistyczny świat
- ❖ Mikroserwisy podbijają serca kolejnych programistów
- ❖ Internet of Things – nikt nie wie co to, ale jest wszędzie
- ❖ Już nie tylko Agile, ale DevOps i Continuous Delivery
- ❖ Dziesiątki fascynujących konferencji IT odbywa się co roku w samej Polsce!

A JUŻ JUTRO...

Nie obudź się z ręką w nocniku!

UCZ SIĘ!

- ❖ Języki funkcyjne (F#, Scala, Elixir, Erlang, Elm, ...)
- ❖ JavaScript i podstawowe frameworki (Angular, React, ...)
- ❖ Programowanie na platformy mobilne
- ❖ Programowanie reaktywne (Rx dostępne na praktycznie każdą platformę)
- ❖ DevOps, Continuous Deployment i chmury (Azure, Amazon), Docker i inne
- ❖ Machine Learning i Data Science

DLACZEGO WARTO? JĘZYKI FUNKCYJNE

- ❖ Podejście funkcyjne umożliwia pisanie prostszego, bardziej zwięzłego i czytelnego kodu – im mniej kodu i łatwiej go czytać, tym mniejsze prawdopodobieństwo popełnienia błędu
- ❖ Skalowalność – w pełni funkcyjny kod jest łatwo skalowalny dzięki zminimalizowaniu efektów ubocznych i współdzielonego stanu (polecam zapoznać się z biblioteką MBrace)
- ❖ Poprawny kod funkcyjny spełnia większość wytycznych dotyczących tworzenia dobrego kodu obiektowego (np. SOLID – funkcje wyższego rzędu, funkcje jako interfejsy)
- ❖ It's fun!

DLACZEGO WARTO? JAVASCRIPT

- ❖ JavaScript jest podstawą Internetu, a Internet jest podstawą naszego życia – warto wiedzieć jak to działa
- ❖ Ogrom powstających narzędzi umożliwia testowanie różnych konceptów programistycznych i „podkradanie” ciekawych rozwiązań
- ❖ Twój aktualny lub następny projekt będzie wymagał znajomości JavaScript
- ❖ JavaScript staje się w pewnym sensie nowym assemblerem – coraz więcej narzędzi pozwala kompilowanie innych języków do JS (patrz TypeScript, Elm, FunScript i inne)
- ❖ W JavaScript powstają już nie tylko aplikacje webowe, ale również serwerowe (Node) i desktopowe (Electron, React Native)

DLACZEGO WARTO? PLATFORMY MOBILNE

- ❖ Smartfony podbiły świat – w tej chwili więcej osób posiada smartfon niż komputer domowy
- ❖ Smartfony są podstawowym narzędziem służącym do dostępu do informacji, w tym Internetu
- ❖ Natywne aplikacje na smartfony czy tablety pozwalają na wykorzystanie pełni możliwości urządzeń – zaawansowanych sensorów, rozbudowanych API
- ❖ Kurczący się rynek PC i rosnący rynek mobilny sprawiają, że Twoje następne projekty będą przynajmniej w części związane z rozwiązaniami mobilnymi

DLACZEGO WARTO? PROGRAMOWANIE REAKTYWNE

- ❖ Żyjemy w asynchronicznym świecie – wszystko opiera się na obietnicy otrzymania odpowiedzi („We’ll call you later”)
- ❖ Nasze aplikacje w coraz większej części będą kompozycją wielu asynchronicznych usług
- ❖ Podejście reaktywne ułatwia nam informowanie konsumenta o asynchroniczności naszego rozwiązania
- ❖ Programowanie reaktywne w językach obiektowych pozwala nam korzystać z wielu zalet programowania funkcyjnego bez konieczności przekonania wszystkich do zmiany języka
- ❖ W stosunku do innych rozwiązań ułatwia składanie ze sobą i testowanie kodu

DLACZEGO WARTO? DEVOPS, CD, CLOUD

- ❖ Budujemy coraz bardziej rozbudowane aplikacje – coraz częściej modułarne i niezależnie od siebie działające; przetestowanie i odpalenie kodu lokalnie staje się coraz trudniejsze i mniej efektywne
- ❖ Konieczne jest ciągłe deployowanie kodu na środowiska testowe gdzie można sprawdzić działanie komponentu w połączeniu z resztą systemu
- ❖ Świadomość środowiska w którym będzie funkcjonował nasz kod ułatwia nam uniknięcie sporej liczby błędów; znajomość zasad działania naszego środowiska i wiedza jak go konfigurować przyspiesza rozwiązywanie ewentualnych problemów i zwiększa poczucie odpowiedzialności za system jako całość
- ❖ Większość nowych aplikacji będzie powstawać z myślą o rozwiązaniach chmurowych, warto wiedzieć jak ten temat ugryźć i jak wykorzystać moc chmury

DLACZEGO WARTO? ML I DATA SCIENCE

- ❖ Nasze systemy muszą radzić sobie z coraz większą ilością danych, których nie jesteśmy już w stanie przetwarzać w czasie rzeczywistym
- ❖ Potrzebujemy narzędzi które będą wyciągać wnioski z istniejących danych i podejmować decyzje na ich podstawie
- ❖ Machine Learning już w tej chwili napędza większość narzędzi z których korzystamy (choćby Google, Facebook, Twitter, SwiftKey, systemy bankowe czy narzędzia developerskie)
- ❖ Rośnie zapotrzebowanie na ludzi którzy będą potrafili wspomóc biznes czy naukowców w wyciąganiu poprawnych wniosków i podejmowaniu decyzji na podstawie dostępnych danych

TO OD CZEGO ZACZAĆ? JAKAŚ PROPOZYCJA?

- ❖ Wybierz sobie dwa tematy – jeden związany z Twoją aktualną pracą, a drugi z goła odmienny (ale taki, który uważasz za interesujący)
- ❖ Zapisz się na jeden z licznych kursów online dostępnych na serwisach takich jak Coursera, NovoEd, Edx, Udacity, MVA i innych (po jednym dla każdego z wybranych przez Ciebie tematów)
- ❖ Przeznacz 2-3 wieczory w tygodniu na naukę nowych rzeczy
- ❖ Zacznij korzystać z Twittera jako narzędzia do przeglądania newsów ze świata IT – obserwuj twórców i osoby powiązane z interesującymi Cię projektami
- ❖ Weź udział w projektach open source – możesz pomóc przy kodzie, dokumentacji, testowaniu
- ❖ Ciągłe poszukiwanie inspiracji – np. weź udział przynajmniej w jednej większej konferencji programistycznej, obejrzyj materiały wideo z innych konferencji (np. NDC, GOTO, CraftConf)
- ❖ Rozmawiaj, dyskutuj, kwestionuj, zadawaj pytania – **i przybывaj na kolejne spotkania!**

linkedin.com/in/bartsokol 

bart-sokol.info 

[@bartsokol](https://twitter.com/bartsokol) 

THE END.

Dziękuję za uwagę!