evropský sociální fond v ČR

EVROPSKÁ UNIE

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1

Šablona: **Inovace a zkvalitnění výuky prostřednictvím ICT**

Název: **Vytváření aplikací pro systém Windows**

Téma: **Principy rychlého vývoje aplikací (RAD)**

Autor: **Ing. Hodál Jaroslav, Ph.D.**

Číslo: **VY_32_INOVACE_27–03**

Anotace: *Materiál představuje principy rychlého vývoje aplikací (RAD) v moderních programovacích prostředích (např. Delphi).*

Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno červenec 2013.

Podpora digitalizace a využití ICT na SPŠ CZ.1.07/1.5.00/34.0632

3. Principy rychlého vývoje aplikací (RAD)

- s pokrokem ve vývoji operačních systémů i uživatelských programů došlo k výrazné změně i v aplikacích, které slouží programátorům k vytváření dalších aplikací
- všechny základní problémy a nástroje byly vyřešeny a hotovy (editor kódu, překladač, ladící a podpůrné nástroje), a proto přišel čas myslet i na pohodlí programátorů
- v roce 1995 přišla společnost Borland s revolučním vývojovým prostředím nazvaným **Delphi**, které poprvé umožnilo pohodlně a rychle sestavit aplikaci pomocí předchystaných součástí (tzv. **komponent**) s využitím velmi uživatelského grafického rozhraní (**IDE**, viz 27-02)

RAD

- pro způsob vytváření aplikací pomocí hotových modulů a komponent se rychle vžil název RAD (**rapid application development**, rychlý vývoj aplikací)
- nástroje RAD se hodí především pro vytváření programů s grafickým uživatelským rozhraním (GUI, dnes ho má drtivá většina programů)
- umožňují rychle sestavit tzv. frontend aplikace, což je právě **rozhraní pro komunikaci** s uživatelem **a obsluhu programu**, a nechávají tak více času pro ladění samotného výkonného kódu, který aplikaci dodává schopnosti
- v oblasti přípravy uživatelského rozhraní výsledné aplikace se potřebný čas zkrátil z mnoha hodin na řádově **minuty**

RAD (2)

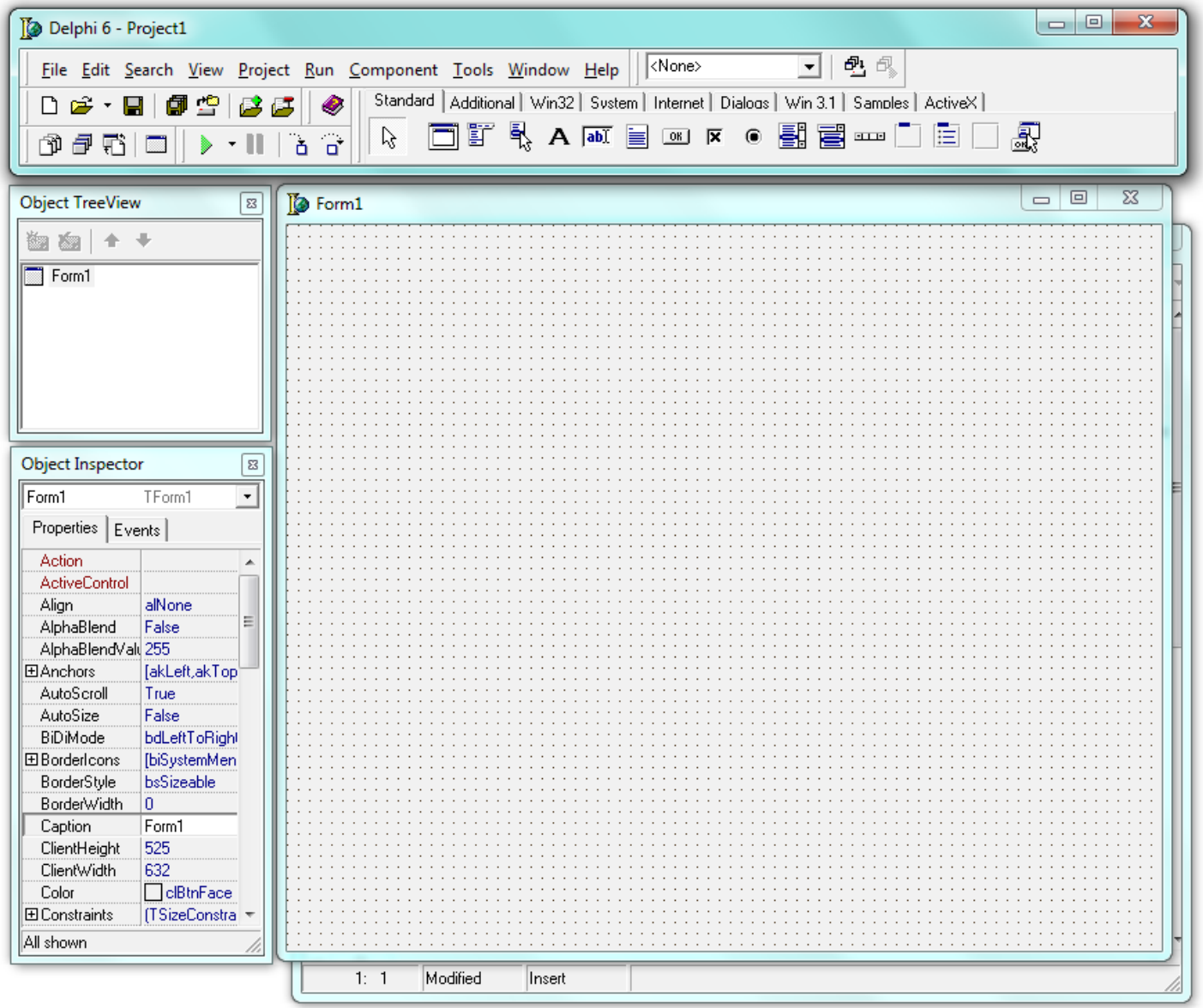
- samotné zrychlení a zjednodušení práce programátora je umožněno několika nástroji
  - předchystaný prototyp aplikace
  - hotové standardizované ovládací prvky
  - inteligentní doplňování kódu

- velká část těchto zjednodušení má svůj základ v důsledném využití **objektového** pojetí programování (viz 26-20 – Objektově orientované programování)
- existuje samozřejmě celá řada dalších nástrojů pro podporu programování, ty se však již mohou lišit v závislosti na použitém programovacím jazyce a vývojovém prostředí

Předchystaný prototyp aplikace

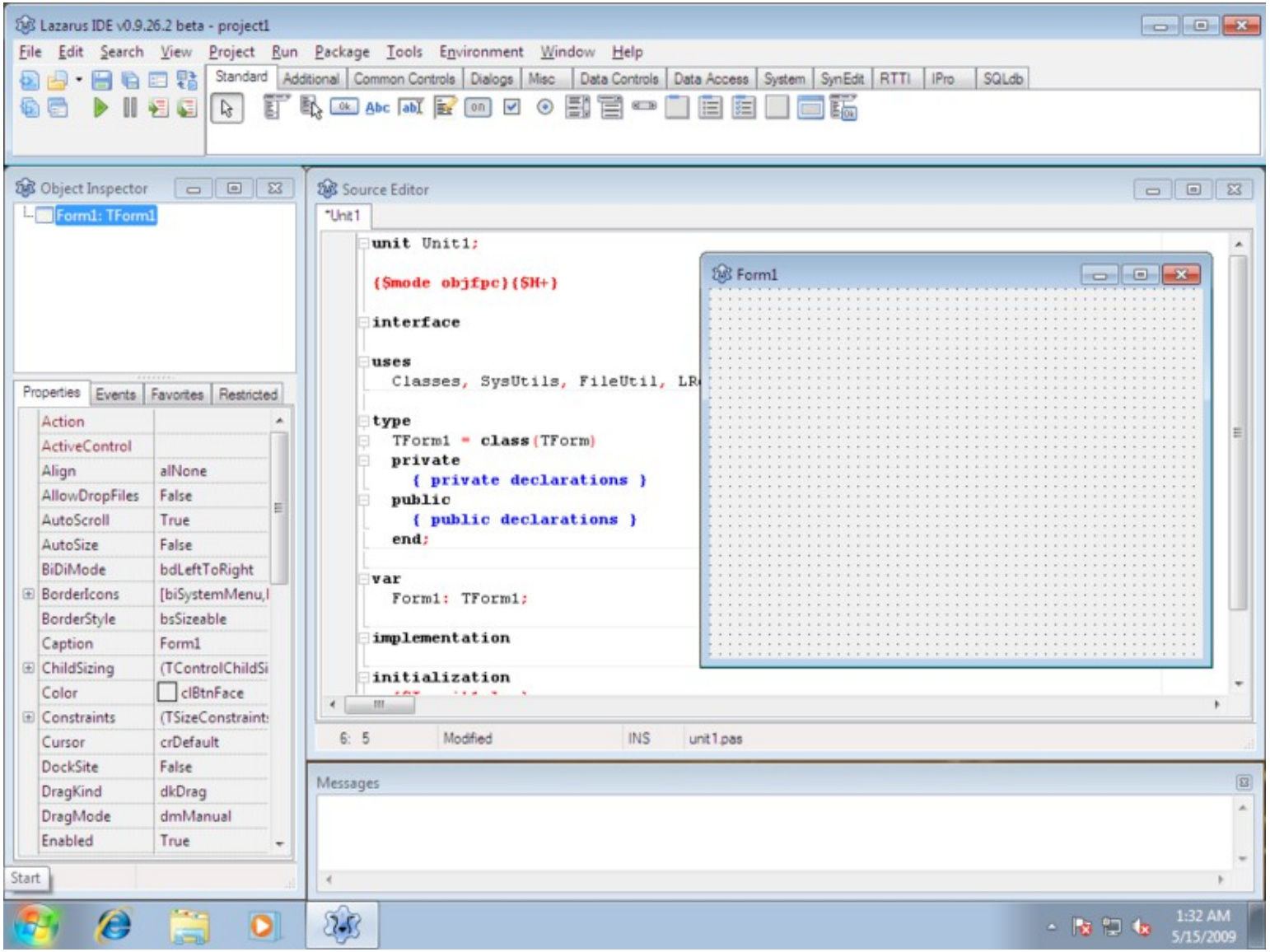
- protože se předpokládá, že téměř každá aplikace určená pro operační systém s GUI bude využívat tzv. **okna**, bude mít každá vyvíjená aplikace minimálně jedno
- toto okno je považováno za **hlavní okno aplikace** (jeho zavřením se obvykle chod aplikace ukončí)
- právě toto okno je hlavní součástí prototypu aplikace
- obsahuje všechny potřebné **metody** a **vlastnosti**, které umožňují ovládat jeho chování i vzhled
- do tohoto okna lze umísťovat rozšiřující ovládací prvky (tzv. **komponenty**)
- součástí prototypu je také základní kód aplikace a zdrojový kód programové jednotky (viz 26-19) spojené s oknem

Delphi



vizuální prototyp aplikace v prostředí Delphi 6 (Borland)

Lazarus



snímek vývojového prostředí Lazarus (dostupný z lazarus.freepascal.org)