

A co je to vůbec ten CAD?

- Computer Aided Design
 - Volně lze přeložit jako **Počítačem podporované konstruování**
- 2D kreslení
 - Ve výrobním průmyslu a strojírenství
 - Historicky
 - Výkresová dokumentace
 - V ostatních odvětvích
 - Schémata, mapy, GIS, ...
- 3D modelování
 - Modelování reality, příprava pro výpočty a výrobu
 - Vizualizace, ...

2D versus 3D

- **2D kreslení** – podpora výkresové dokumentace
 - „příjemné“ elektronické kreslicí prkno
 - snadné změnové řízení
 - přesnost grafických metod na úrovni analytických
- **3D modelování** – modelování reality „1:1“
 - objemové
 - nejbližší realitě
 - plošné
 - zpravidla tam, kde objemové modelování „nestačí“
 - hybridní
 - kombinace objemového a plošného – nabízí Inventor
- **3D parametrické modelování** – „ladění“
 - vytváření variant
 - možnost optimalizace

Výkresová dokumentace a 3D model

- Výkresová dokumentace – stále nejčastější výstup
 - CAM a/nebo Rapid Prototyping nelze aplikovat vždy
 - Vazba CAD – výroba často (nejčastěji) lidským faktorem
- Výkresy se odvozují od 3D modelu
 - Snaha o maximální automatizaci
 - Automatizovat však zřejmě nikdy nepůjde 100%
 - Snaha o obousměrnou vazbu mezi výkresem a modelem (např. změna rozměru při změně kóty)

Vizualizace vs. 3D

- Vizualizace vychází z 3D modelu
- Rozšíření modelu
 - Povrchy
 - Osvětlení
 - Pohledy
 - Zasazení do okolí (nemusí být plně 3D – např. zasazení domu do fotografie stávajícího stavu, nebo návrhu auta na parkoviště plné konkurenčních vozů)
 - Předměty „běžného života“
 - Chodci, auta, oblaka, zařízení/„nepořádek“, animované prvky (krb, televize)

CAD-systémy jejichž jméno „má zvuk“

- CATIA (Dassault Systems)
 - Nejrozšířenější CAD automobilového průmyslu
- Pro/Engineer (Parametric Technology)
 - Nejrozšířenější CAD ve strojírenství
- I-DEAS
- Euclid Marta (Datavision)
- Autodesk Inventor (Autodesk)
- AutoCAD (Autodesk)
 - Nejrozšířenější 2D CAD-systém
- CADDS5 (Computervision)
- Unigraphics (GM EDS)
- Solid Designer (Hewlett-Packard)
- I/EMS (Intergraph)

- Nemetschek
 - Nejrozšířenější evropský CAD ve stavebnictví a architektuře

AutoCAD

- Výkresová databáze
 - Grafický náhled – pouze prezentační vrstva
 - Přesné kreslení – systém speciálních DB-dotazů
- Základní typy entit
 - Křivka, úsečka, oblouk, text, kóta, ...
 - Nemají jasnou sémantiku
- Systém hladin
- Silná vazba na výstupní výkres
- 3D jako „dodatek“
 - Bez jasné sémantiky
 - V kolizi s metodami pro 2D kreslení

Autodesk Inventor

- Parametrický modelář
 - 3D modelování
 - Parametrizace
 - Hybridní modelování – objemové + plošné
- Konstrukční prvky neboli *features*
 - Sémantika prvků
- Historie geneze modelu
- Primitivní kinematika
- Odvozování výkresové dokumentace

CAM versus Rapid Prototyping

- Computer Aided Manufacturing
 - Výroba skutečných součástí
 - Výroba prototypů skutečných vlastností
 - Malosériová výroba
 - ...
- Rapid Prototyping
 - Výroba tvarově korektních modelů
 - Specifická technologie výroby – „3D tisk“, vytvrzování, ...
 - Modely nemají požadované fyzikální vlastnosti
 - Používají se pro rozměrovou představu (zástavba motoru, ...)
 - Možnost použití ve slévárenství – modely

CAE – Computer Aided Engineering

- Výpočetní moduly
- Kinematika
- Dynamika
- FEM – Finite Element Method, neboli Metoda konečných prvků
 - Využívá se pro analýzu šíření polí v kontinuu
 - Napjatostní analýzy
 - „Crash-testy“

Správa dat ve výrobní sféře

IS – Enterprise Information System (*podnikový informační systém*)

DW – Data Warehouse (*datový sklad*)

DW p – Data Warehouse pump (*pumpa datového skladu*)

PDM – Product Data Management

PLM – Product Lifecycle Management

ERP/SCM – Enterprise Resource Planning / Supply Chain Management

CRM – Customer Relationship Management

EIP – Enterprise Information Portal

EAP – Enterprise Application Portal

Motto: CAD-systém je databáze

- CAD-systémy jsou velmi specifické databáze
 - Specifická reprezentace: grafická
 - Specifický DDL (Data Definition Language)
 - Pevná množina entitních typů (v AutoCADu ne zcela platí)
 - Rozšiřování entit
 - Sémantická pravidla (definice a používání hladin, ...)
 - Specifický DML (Data Manipulation Language)
 - Operace typu _LINE, _MOVE, ...
 - Specifický dotazovací jazyk
 - Operace typu _DIST, _PER, _MID, IntersectWith, ...
 - „Více databázový“ pohled nabízejí různé výběry objektů s možností filtrace dle rozličných parametrů (obdoba klausule WHERE v SQL)
- Přesné kreslení
 - Dotazy nad databází
 - Analytická geometrie
 - Více při praktických cvičeních

Databáze CAD-systému na příkladu databáze AutoCADu

Základní prvky databáze

- Kontejnery
 - Mapování jména symbolu (textový řetězec) na databázový objekt (ID, handle, seznam, ukazatel, ...)
 - Dva druhy kontejnerů
 - Tabulky symbolů
 - Slovníky
- Entity
 - Speciální objekty v grafické databázi – s vizuální reprezentací ve výkresu AutoCADu
 - Obsaženy ve speciálním kontejneru – Block Table – jakožto Block Table Records

Kontejnery – tabulky symbolů

- Pevně daný počet – 9 tabulek
- Tabulky mívají iniciální záznamy, které jsou přítomné vždy, např. hladina 0 nebo modelový a výkresový prostor, atd.
- Každá tabulka obsahuje specifickou třídu záznamů
 - Block Table
 - *MODEL_SPACE
 - *PAPER_SPACE
 - Layer Table
 - Linetype Table
 - Dimension Style Table
 - Registered Applications Table
 - Text Style Table
 - User Coordinate System Table
 - Viewport Table
 - View Table

Kontejnery – slovníky

- Obecnější než tabulky symbolů
- Mohou obsahovat libovolné potomky základního typu objektu z objektového modelu AutoCADu
- Slovník pojmenovaných objektů
 - „Slovník slovníků“, tj. kořenový slovník, přes který jsou přístupné další slovníky
 - Je generován s novým výkresem (tj. prázdnou databází, resp. s databází v iniciálním stavu)
- Lze volně přidávat vlastní slovníky
 - Úložiště negrafických aplikačních dat
 - Jsou dosažitelné ze slovníku pojmenovaných objektů

Specifická (aplikační) data objektů

- Obdoba uživatelských slovníků
- Vazba ke konkrétním objektům (entitám)
- Rozšířená data - Extended Data
 - Seznam záznamů ze základních typů AutoCADu
 - Různé seznamy pro různé aplikace (řídí se jménem aplikace)

- Omezená kapacita
- Rozšiřující slovníky – Extension Dictionaries
 - Flexibilnější a méně omezující obdoba rozšířených dat
 - Snáze se sdílí mezi aplikacemi
 - Oproti rozšířeným datům je složitější režie

Entity

- Základní
 - 1 entita = 1 záznam v Block Table
 - Specifickou množinou jsou křivky (obecně; ne ve významu AcDbPolyLine, tj. výsledku příkazu KŘIVKA)
 - Podobně i 3D tělesa
- Složené
 - 1 entita = více záznamů v Block Table
 - Entita je de facto vnořeným kontejnerem pro další (speciální) entity

Základní vlastnosti entit

- Barva (Color)
 - Číslo barvy (pevná paleta, 256 odstínů)
 - Speciální barvy – BYBLOCK, BYLAYER, ...
- Typ čáry (Linetype)
- „TČMĚŘ“ – volně asi Měřítko přerušování čáry (Linetype Scale)
- Viditelnost (Entity Visibility)
- Příslušnost do hladiny (Entity Layer)

Základní funkce pro entity (resp. metody entit)

- **Pozn.:** Částečně závislé na rozhraní
- Výpočet průsečíku
- Transformace (transformační maticí) a projekce
- „klíčové body“ – Osnap, Grip and Stretch points
- Vykreslování
- Výpis (informační výpis – _LIST)
- Geometrické meze
- Rozložení na primitiva
- Práce s vnořenými entitami

Vlastní entitní typy (systém je otevřený)

- Mechanizmy OOP, dědičnost, ...
- Vychází se z některé úrovně objektového modelu
 - Od obecné reprezentace entity (AcDbEntity)
 - Přes např. obecnou křivku (AcDbCurve)
 - Až ke konkrétnímu entitnímu typu (AcDbLine)
- Nutnost řešení zobrazování „kdykoliv“
 - Servisní aplikace
 - Zástupné PROXY objekty

Práce s databází

- Velmi závislá na vývojovém prostředí
- Více strategií přístupů k objektům (Zejména u nízkoúrovňových prostředí)
 - Otevírání a zavírání objektů
 - Transakční zpracování
- Některá prostředí umožňují využívat událostní model databázového stroje AutoCADu

Objektové identifikátory

- AcDbObjectId, ads_name
 - Dvě různé, ale navzájem převoditelné reprezentace
 - Jedinečný v seanci AutoCADu (i v rámci MDI/MDE)
 - Nepřetrvává při uložení a načtení výkresu
- AcDbHandle
 - Jedinečný ve výkrese (v MDI/MDE možné kolize mezi databázemi)
 - Přetrvává při uložení a načtení výkresu

Databáze AutoCADu vs. Dokument Windows

- Dokument Windows je v AutoCADu DWG (**drawing**)
 - DWG je obrazem konkrétně naplněné databáze
- Jedno versus více- dokumentové rozhraní
 - Staré verze AutoCADu – SDI (Single Document Interface)
 - Aktuální verze AutoCADu – MDI (Multiple Document Interface), neboli MDE (Multiple Document Environment)
- 1 Dokument ≈ 1 Databáze

Vývojová prostředí: AutoCAD

Úhel pohledu

- AutoCAD jako otevřený systém
 - Přehled vývojových prostředí
 - Porovnání a možnosti
- ACAD jako jádro profesně orientovaných aplikací
 - MDT, ADT – představení
 - Role vývojových prostředí
 - Další – profesně specifická – vývojová prostředí

AutoCAD jako otevřený systém

- Nejotevřenější systém CAD
- Možnost hlubokého a prorostlého zahrnutí aplikací
- Datová otevřenost - totální
 - Přístupná grafická databáze pro čtení i zápis
 - Možnost vlastního datového rozšíření entit
 - Možnost vytváření vlastních typů entit (s využitím objektového rysu dědičnosti)
- Funkční otevřenost - velká
 - Přístupná široká (avšak omezená) škála funkcí ACADu
- Otevřenost na úrovni uživatelského rozhraní

Co aplikační rozhraní zpřístupňují?

- Uživatelské rozhraní
 - Menu, nástrojové lišty, ...
 - Definice vlastních příkazů
 - Vstup/výstup na příkazové řádce
 - Dialogové panely
- Grafická databáze – „výkres“
 - Entity výkresu
 - Vlastnosti entit
 - Geometrické informace a vztahy mezi entitami
- Systémové prostředí AutoCADu (nastavení)
 - Nastavení hladin, stylů, ...
 - Nastavení prostředí (zobrazení, uchopování, ...)

Co uživatelská rozhraní zpřístupňují?

- Datová rozšíření
 - Uživatelské slovníky
 - Datová rozšíření entit
 - Vlastní entitní typy
- Reakce na události – tzv. reaktory
 - Reaktory MDE (např. přepnutí mezi dokumenty)
 - Reaktory editoru (např. začátek / konec příkazu)
 - Změny entit
 - Změny v databázi
 - Transakční reaktory
 - ...

Vývojová prostředí AutoCADu – přehled

- Funkční a datová rozhraní
 - AutoLISP / VisualLISP
 - C
 - ADS ADSRX (†)

- C++
 - ObjectARX ObjectDBX
- DIESEL
- VBA
- ActiveX a COM
- .Net
 - ObjectARX Managed Wrapper Classes
- Uživatelská rozhraní (pokročilá)
 - Jazyk systému menu
 - DCL (+)
 - MFC a Win32
 - ObjectARX MFC Extension
 - nativní prostředky použitých jazyků [VB(A), Java, Delphi, ...]
 - .NET WinForms

Porovnání funkčních a datových rozhraní

- AutoLISP / VisualLISP
 - snadné programování
 - malá rychlost
 - omezená hloubka záběru
 - z historie (není již favorizován)
- C (ADS, ADSRX +)
 - rozšíření AutoLISPU oproti LISPU převedená do C (vyšší rychlost, nepříjemné použití)
- C++ (ObjectARX, ObjectDBX)
 - nejhlubší záběr (nové typy entit)
 - maximální výkon
 - minimální produktivita
- DIESEL
 - pro specifické využití
- VBA
 - leží mezi LISPEM a C++
 - proti C++ rychlost ↓, produktivita ↑
- Prvky ActiveX, sada rozhraní COM
 - otevření dalším jazykům
- .Net – ObjectARX Managed Wrapper Classes
 - obal ObjectARX
 - t.č. pouze podmnožina (neobsahuje DBX)
 - velká jazyková škála
 - Garbage Collection
 - produktivitou mezi VBA a C++

Porovnání pokročilých uživatelských rozhraní

- Jazyk systému menu
 - specifické využití
- DCL (+)
 - neproduktivní (textový zápis)
 - omezený
- MFC a Win32
 - problematické
 - „všemocné“
- ObjectARX MFC Extension
 - rozšíření MFC o standardní prvky ACADu
- Prvky ActiveX, rozhraní COM
 - některé standardní prvky AutoCADu
 - zpřístupnění nových funkcí typu ADC, okno vlastností, ...
- .Net (WinForms)
 - pravděpodobná cesta budoucího vývoje AutoCADu

Vztah AutoCADu a Windows

- AutoCAD pouze pro platformu MS Windows, vazba je velmi těsná
- Vystavěn prostřednictvím Win32 a MFC (a ActiveX/COM)
 - Problémy
 - MDI (resp. MDE)
 - události typu „přepnutí mezi dokumenty“, zavření dokumentu
 - stav bez dokumentu (tj. bez grafické databáze)
 - globální proměnné (globální vs. „per document“)
 - Možná kolize verzí MFC při dynamickém linkování
 - Řízení zdrojů (Windows resources) při dynamickém linkování
 - sdílení jeden prostor
 - přepínání zdrojů mezi zdroji zahrnuté aplikace a AutoCADu
- Windows přechází na .Net jako hlavní platformu
 - WinForms – nahrazuje MFC
 - C# je deklarován jako hlavní systémový jazyk
 - Lze předpokládat následování AutoCADem

AutoCAD - jádro profesně orientovaných aplikací

- Obecný AutoCAD
 - příliš volnosti
 - hladiny bez jednotného systému používání
 - sada základních entitních typů bez určeného vztahu k realitě
 - obecné 3D a obecná (strojařská?) tvorba 2D dokumentace
 - ...
 - chybějí specifické funkce, knihovny součástí a materiálů, ...
 - široká řada „amatérských“ nastaveb
- AutoCAD se ve strojírenství a později i ve stavebnictví stává jádrem profesně orientovaných aplikací

Autodesk Mechanical Desktop – MDT

- Oblast využití – strojírenství
 - Základ 3D modelování
 - 2D výkresy – „odpadní produkt“
 - Parametrický (!) modelář
 - Parametrizace vazbami a kótováním (snižování st. volnosti)
 - Systém proměnných a výrazů
 - Vazba na MS Excel – tvorba variant
- V současné době je nahrazován produktem Autodesk Inventor (není na bázi AutoCADu)

Autodesk Architectural Desktop – ADT

- Oblast využití – stavebnictví
 - Základ 3D modelování
 - Souběžná existence 2D reprezentace (různé pohledy)
 - Množství objektů specifických pro stavebnictví (AEC)
 - stěny, dveře, okna, schodiště, ...
 - Jiný způsob práce (skládání „prefabrikátů“)

Role vývojových prostředí

- MDT a ADT jsou realizovány převážně pomocí ObjectARX/DBX, jazyce menu a MFC/Win32
 - Tj. jsou vytvořeny těmi nástroji, které má každý vývojář volně k dispozici
- Další – profesně orientovaná – prostředí
 - Nově vzniklé vrstvy rozhraní pro xDT jsou formou knihoven rozšiřujících rozhraní ACADu zveřejněny
 - Zpřístupňují specifickou funkčnost a objekty
 - Obalují některé části ADS, ObjectARX/DBX a MFC další „pohodlnější“ vrstvou
 - Poskytují další rozšíření MFC – MFC Extension se standardními prvky daného DT

Role ObjectDBX – Database Extension

- Umožňuje definici vlastních typů entit (AEC_DOOR ~ dveře)
- Typ se vytváří děděním z požadované úrovně
 - AcDbEntity – nejnížší smysluplná úroveň („prázdný“ typ s abstraktními virtuálními metodami)
- Nutno řešit veškerou funkcionalitu

- Ukládání / načítání do DWG a DXF
- Zobrazování
- Rozklad na primitivní typy (nejen pro EXPLODE)
- Chování vzhledem ke standardním editačním postupům
- Geometrické dotazy (průsečíky, derivace v bodech, ...)
- Bez přítomnosti servisního DBX – PROXY
 - Možnost dvojího zázemí – plné / tzv. object enabler

Inventor – vývojové prostředí

Inventor API

- Jediné aplikační rozhraní
- Založeno na COM
- Jazyky
 - Základní (podporovaný Autodeskem): VBA
 - Možno použít jazyky podporující COM
 - Možno použít jazyky podporující .Net
 - Problém se silnými jmény
 - „Kuchařka“ pro C# by se měla objevit v průběhu semestru

Obecně

Uplatnění vývojových prostředí

- *Interní vývoj CAD-systémů*
- Přizpůsobování podnikovým procesům
 - Návrhovým/vývojovým
 - Organizačním
- Integrace softwarových systémů
 - Integrace vývojových procesů
 - Vazba na výpočetní moduly (CAE)
 - Časté uplatnění – výpočetní systémy založené na metodě konečných prvků (FEM)
 - Vazba na CAM
 - Rychlá výroba modelů, výroba prototypů, kusová výroba finálních produktů
 - Datová integrace v rámci podnikového prostředí
 - Potlačení izolace CAD
 - Integrace dat do Enterprise IS
 - Správa a sdílení dat (PDM), podpora týmové práce
 - Řešení dodavatelsko-odběratelských vztahů (CRM)
 - Vazba na Internet (zejména webové portály)

Základy programování pro 2D CAD

Visual Basic

Ostatní prostředí představená minulý týden jsou mimo rozsah předmětu

Syntaxe jazyka I

- Řádková orientace, oddělovače příkazů
 - Konec řádku ≈ oddělovač příkazů
 - Více příkazů na řádce – odděleno dvojtečkou :
 - Jeden příkaz na více řádek – spojeno podtržítkem na konci řádku
-
- Tečková notace
 - Oddělování kvalifikátorů tečkou
YourProject.YourModule.YourSub MyProject.MyModule.MyVar
 - Objektový jazyk? – do verze .Net nikoliv; jisté rysy ano
- Kometáře
 - Řádkové – příkaz **Rem** nebo apostrof ' ,

Syntaxe jazyka II

- Výrazy
 - Infixová notace L op P
 - Přiřazení – rovnítko =
 - $a = 1 + 2$
 - Porovnání – rovnítko =
 - If a = 3 Then MsgBox "Trojka!"
 - Logické operace – klíčová slova **And, Or, Not, ...**

- Proměnné a konstanty
 - Deklarace – kdekoliv v kódu
 - **Dim *proměnná* As *typ***
 - **Const *konstanta* As *typ* = hodnota**

Datové typy

- Základní
 - Currency
 - Double
 - Integer
 - Single
 - String
 - Variant
 - Object
 - Decimal
 - Array
- Uživatelem definované

Type MyType

 MyName As String MyBirthDate As Date

MySex As Integer

End Type

- Pole

Dim MyArray(10, 10) As Integer

MyArray(7, 5) = 8

- ReDim pro realokaci
- Možno *od to do*

Příkaz With

With Worksheets("Sheet1").Range("A1:C10")

 .Value = 30

 .Font.Bold = True

 .Interior.Color = RGB(255, 255, 0)

End With

Procedury

- Sub Procedures – nevrací výsledek

Sub *jméno* (*parametry*)

 ' ... příkazy ...

End Sub

- Function Procedures – vrací výsledek

Function *jménoFunkce*(*parametry*)

 ' ... příkazy ...

jménoFunkce = *výsledek*

End Function

Vlastnosti

- Nastavování a získávání hodnot
 - Hodnoty nemusí být přímo skladované – možnost odvození výpočtem
- Nastavování

Property Let Names(*p1* As *typ*, *p2* As *typ*, ...)

 ' ... příkazy ...

End Property

- Získávání

Property Get Names(*p1* As *typ*, *p2* As *typ*) As *typVýsledku*

 ' ... příkazy ...

End Property

Podmíněné příkazy

- If – Then – Else

If performance = 1 Then

 Bonus = salary * 0.1

```

Elseif performance = 2 Then
    Bonus = salary * 0.09
Elseif performance = 3 Then
    Bonus = salary * 0.07
Else
    Bonus = 0
End If

```

- Select Case

```

Select Case performance
Case 1
    Bonus = salary * 0.1
Case 2, 3
    Bonus = salary * 0.09
Case 4 To 6
    Bonus = salary * 0.07
Case Is > 8
    Bonus = 100
Case Else
    Bonus = 0
End Select

```

Cykly

- Do – Loop (While | Until)

' S podmínkou na začátku

```

Do Until myNum = 10
    myNum = myNum - 1
    counter = counter + 1

```

Loop

' S podmínkou na konci

```

Do
    myNum = myNum - 1
    counter = counter + 1

```

Loop While myNum > 10

' S ukončením „uprostřed“

```

Do Until myNum = 10
    myNum = myNum - 1
    counter = counter + 1
    If myNum < 10 Then Exit Do

```

Loop

- For – Next

```

For j = 2 To 10 Step 2
    total = total + j
Next j

```

- For Each – Next

```

For Each I In TestArray
    TestArray(I) = I
Next I

```

Formuláře

- Vizuální editor

Visual Basic v AutoCADu

- Vývojové prostředí VBA v AutoCADu
 - Příkaz `_VBAIDE`
 - Menu: Nástroje -> Makro -> ...
 - Nápověda
 - Menu: Nápověda -> Nápověda pro vývojáře
 - Témata nápovědy:
 - ActiveX and VBA Developer's Guide
 - ActiveX and VBA Reference

Document.Utility – uživatelský vstup

- Metody Get...
 - GetAngle = vrací úhel
 - GetCorner = vrací roh
 - GetDistance
 - GetEntity
 - GetInput
 - GetInteger
 - GetKeyword
 - GetOrientation
 - GetPoint
 - GetReal
 - GetRemoteFile
 - GetString
 - GetSubEntity

Přístup k databázi

- Modelový prostor aktuální databáze
 - ThisDrawing.ModelSpace
 - Vkládání entit do databáze
 - ThisDrawing.ModelSpace.AddLine

Příklad

Sub test()

Dim startPt As Variant, endPt As Variant

startPt = ThisDrawing.Utility.GetPoint(, "První bod: ")

endPt = ThisDrawing.Utility.GetPoint(startPt, "Druhý bod: ")

'Variant nyní zastupuje trojrozměrné pole doubles

Dim newLine As AcadLine

Set newLine = ThisDrawing.ModelSpace.AddLine(startPt, endPt)

End Sub

Uživatelské rozhraní ve VBA v AutoCADu

- Formuláře
- Grafický editor
 - Definice vzhledu
 - Definice událostí
- Interakce s AutoCADem
 - Zobrazení formuláře
 - *UživatelskýFormulář.Show*
 - Další interakce při ošetřování událostí

Události vs. reaktory

- Ve VBA správa událostí ≈ reaktory v jiných vývojových prostředích
- Vytvoření třídy EventClassModule
 - Public WithEvents App As AcadApplication
 - Analogické třídy pro ostatní typy událostí
- Připojení k objektu generujícím události
 - Dim X As New EventClassModule
 - Sub InitializeEvents()
 - Set X.App = ThisDrawing.Application
 - End Sub
 - Call InitializeEvents