
1. Charakteristika disciplíny SW inženýrství a její vývoj

Přednáška: 1

Skripta: ZSWI - OOpřístup .9 -11

1.1. Definice

Disciplína počítačové vědy zaměřená na vývoj velkých SW systémů

Zahrnuje:

- technologické aspekty vytváření SW systémů (modelování, implementace, testování)
- aspekty řízení (vedení týmů, plánování)

1.2. Vývoj

- vývoj v nízkourovňových jazycích,
- programování v jazycích 3.generace (COBOL, PL/1, RPG),
- koncem 60.let - strukturované programování - vede k rozvoji strukturovaných metod analýzy a designu – počátek SW inženýrství
- objektově orientované programování a OO analýza a design - nastupují novou éru softwarového inženýrství

Procedurální přístup - data a funkce odděleny

OO přístup - data a procedury jsou spojeny v objektech

2. Algoritmus

Skripta: ZSWI - ZaklTémata kap.3

(definice a vlastnosti algoritmu, způsoby vyjádření algoritmu, příklady algoritmů)

2.1. Definice

Každý přesný popis, jímž je jednoznačně určen postup řešení úlohy pomocí definované soustavy operací. Každý algoritmus, který poskytuje řešení daného problému je správný – ale nemusí být nejefektivnější.

2.2. Vlastnosti

- Diskrétnost – posloupnost nespojitých, oddělených a předem definovaných operací

Rezultativnost – po **konečném** počtu kroků dospěje k výsledku

- Determinovanost – vždy je jednoznačně určena operace, která následuje
- Hromadnost - dojde ke správnému výsledku pro všechny vstupy z množiny definované v zadání

Složitost

- Časová
Log n, n, n*log n, n², n*log n², n³... 2ⁿ
- Paměťová

3. Programování

Skripta: ZSWI - ZaklTémata kap.4

(program, programovací jazyk, generace programovacích jazyků)

3.1. Program

Vyjadřuje posloupnost operací, které má počítač provést a určuje data, se kterými se tyto operace budou provádět.

Program ve strojovém kódu má instrukční a datovou část.

3.2. Programovací jazyk

Abstraktní počítač, který rozumí příkazům daného jazyka. Příkazy programovacího jazyka představují určitý stupeň abstrakce objektů zpracovávaných skutečným strojem.

3.3. Úroveň programování

- Mikrokód (uvnitř procesoru: popisuje kroky prováděné při každé instrukci)
- Strojový kód (instrukce pro procesor)
- Vyšší programovací jazyky
- Programování řízené událostmi
- Komponentový vývoj

3.4. Generace jazyků

- vývoj v nízkourovňových jazycích,
- programování v jazycích 3.generace (COBOL, PL/1, RPG),
- koncem 60.let - strukturované programování - vede k rozvoji strukturovaných metod analýzy a designu – počátek SW inženýrství
- objektově orientované programování a OO analýza a design - nastupují novou éru softwarového inženýrství

Fortran (1954), Cobol (1960), Basic (1965), Pascal (1971), C (1974), Smalltalk (1980), C++ (1986), Turbo Pascal (1990), Visual Basic (1991), Object Pascal (1995), Java (1996), C# (1999)

4. Druhy programovacích jazyků

ZSWI - ZaklTémata kap.4

(kompilované, interpretované, procedurální, neprocedurální, procedurálně, událostmi řízené)

4.1. Procedurální

Popisují postup, jak dojít k výsledku. Posloupnosti příkazů, cykly, větvení.

Java, C++, Pascal, Basic...

4.2. *Neprocedurální*

Pouze popisují, co chceme dostat jako výsledek.
SQL

4.3. *Kompilované*

Program je přeložen do strojového kódu pro daný procesor. Před spuštěním programu.

4.3.1. *kompilace*

- Kontrola syntaxe zdroje, vytvoření stromové reprezentace
- Generování cílového kódu

4.4. *Interpretované*

Program zůstává ve zdrojáku. Při spuštění je prováděn příkaz po příkazu.
Skriptovací jazyky (JS, VBS, PHP, ASP), Basic (interpret i kompilátor)
Java: překlad na byte-kód a potom interpretace

Při interpretaci je možné část kódu předkompilovat. Při kompilaci naopak může být část programu interpretována.

4.5. *Jazyk*

Gramatika

- Typ 0
- Typ 1 – kontextová gramatika
- Typ 2 – bezkontextová gramatika
- Typ 3 – regulární gramatika

5. Návrhové vzory

Přednáška: 6

Návrhové vzory definují ověřená řešení určitých problémů návrhu.

- Programátoři často vytvářejí části programů napodobováním jiných programů - musí pochopit vzor kódu a použít jej
- Knihy o algoritmech popisují různé vzory algoritmů, které jsou efektivní a prověřené (třídící algoritmy)

5.1. *Výhody*

- zvýšení **znovupoužitelnosti a produktivity** díky používání předdefinovaných návrhů, struktury a kódu
- zachycení specifické znalosti kvalifikovaných návrhářů, tak aby nebyla ztracena při jejich odchodu z organizace
- dosažení rychlého, efektivního a standardního vývoje - zaměření vývojářů na to **co**, vytvářet než jak to vytvářet
- snadná implementace nových verzí - automatizovaně podle vzorů
- zvýšení kvality kódu, jelikož každý vzor obsahuje kód, který již byl testován a použit při vývoji.

6. Webové služby

Přednáška: 11

Skripta: ZSWI - ZaklTémata kap.6

Libovolná služba prováděná prostřednictvím rozhraní WSDL
Standardní způsob propojení vzdálených komponent

Postaveno na již přijatých standardech: XML, http

Protokol: SOAP

Adresář dostupných služeb: UDDI

Web Services Description Language: WSDL

7. Charakteristika datové základny

Skripta: ZSWI - ZaklTémata kap.2

(datové typy a datové struktury včetně dynamických datových struktur – seznam, fronta, zásobník, strom)

7.1. Datový typ

Datový objekt představuje data, se kterými program pracuje.

Datový typ určuje množinu hodnot datového objektu a množinu operací, které s ním lze provádět.

- Jména proměnných – identifikátory proměnných
- Jména konstant – identifikátory konstant
- Literály – jejich jména jsou tvořena speciálními lexikálními elementy

Typované proměnné

Efektivní cílový program

Odhalí nedovolené operace už při překladu

Deklarace datového objektu určuje

Druh objektu (proměnná, konstanta)

Typ

Jméno (identifikátor)

Hodnota

Způsob přidělení paměti

Typ určuje operace, které lze provádět s tímto objektem.

7.2. Jednoduché typy

jednoduché, nedělitelné struktury (číslo, znak, boolean...)

7.3. Strukturované typy

Složkami jsou jednoduché proměnné nebo opět strukturované typy

Operace:

Selektory - pole a[i]

Operace s celou strukturou
Vložení nebo odebrání složky
Vytvoření nebo zrušení datové struktury

7.4. Ukazatelové typy

Slouží především pro identifikaci dynamických objektů
Ukazatel je datový objekt, jehož hodnota identifikuje jiný datový objekt (abstrakce adresy)

8. Souborová a databázová koncepce datové základny

Skripta: ZSWI - ZaklTémata kap.2

8.1. Datové struktury

Účelné uspořádání dat v paměti počítače, soubor pravidel a omezení které určují vztahy mezi prvky dat

- Celočíselná proměnná
- Reálná proměnná
- Komplexní proměnná
- Logická proměnná
- Textový řetězec
- Vektor (pole)
- Matice (dvojměrné pole)
- N-rozměrné pole
- Věta
- Soubor
- Seznam
- Fronta (FIFO)
- Zásobník (LIFO)
- Strom
- Síť
- Tabulka (obsahuje různé typy)

8.2. Fyzická realizace dat

- Bit (binary digit)
- Byte
- Slovo (8b, 16b, 32b, 64b)
- Blob (větší množství B)
- Blok (I/O)
- Sektor (na disku)
- Stránka

8.3. Databáze

Nevýhody uspořádání dat v souborech

- Redundance a nekonzistence dat (stejná data na více místech, těžko aktualizovatelná)
- Problémy při transakcích
- Problémy s vícenásobným přístupem
- Bezpečnost

8.3.1. SŘBD

- Query procesor – zpracování dotazů
- Storage manager – řídí ukládání dat

8.3.2. Typy DB systémů

- Hierarchické DB (IMS od IBM)
- Síťové DB
- Relační DB
 - DB se jeví jako množina tabulek
 - Uživatel mezi tabulkami nevidí fyzické vazby (vazby realizovány pomocí cizích klíčů)
 - Je k dispozici neprocedurální jazyk (výběr, úpravy záznamů) SQL
- Objektově orientované DB

9. XML

Přednáška: 11

Skripta: ZSWI - ZaklTémata kap.4

Značkovací jazyk. První byl GML (Generalized Markup Language – IBM 1969), potom SGML, HTML (Tim Brenenser-Lee). Dnes rozmach XML.

XML je strukturovaný textový formát

9.1. Použití

Přenos dat mezi počítači, MT, PDA...

Tvorba WWW: oddělení obsahu a formy

Výměna a uchovávání dat.

9.2. Struktura

```
<?xml version="1.0"?>
```

```
<značky>
```

Elementy – označené části

9.3. Kontrola platnosti - správnosti

9.3.1. DTD

Sada pravidel pro definici struktury dokumentu

V xml dokumentu musí být uveden DOCTYPE

9.3.2. XML Schéma

Alternativní technologie k DTD. Definuje strukturu dokumentů.

9.4. Transformační technologie

XSL (Extensible Stylesheet Language) – transformace XML dokumentu na např. HTML

9.5. Zpracování XML dat

Parsery

- Nevalidující – kontrolují jen správnost XML
- Validující – kontrolují i správnost podle DTD nebo XML Schématu

10. UML

Přednášky: 4,5

Skripta: ZSWI - OOpřístup kap.3

návrh SW aplikace, který předchází kódování

10.1. CRC karty

Kartičky tříd, na jedné straně odpovědnosti na druhé spolupracovníci.

- Odpovědnost (co třída zná nebo dělá)
- Spolupracovník – pokud třída potřebuje informace, které nemá, nebo pomocí spolupracovníka změní informace, ke kterým nemá přímo přístup

10.2. Historie UML – Unified Modeling Language

Mnoho OO metodik, různé notace è firma Rational, vznik UML 1995

Organizace OMG - Object Management Group

slouží pro zachycení výsledků analýzy a návrhu pomocí různých typů diagramů

není to metodika – využívá se ale v řadě metodik

existuje pro něj řada CASE nástrojů

10.3. Diagramy struktury

10.3.1. Diagram tříd (Class diagram)

statický pohled na systém

reprezentuje strukturu tříd v rámci systému

u každé třídy zachycuje atributy a metody, vyjadřuje vztahy mezi třídami.

Konvence

- Jméno třídy tučně, vycentrovat
- Atributy a metody s malým písmenem, zarovnat vlevo
- Jména abstraktních tříd a metod kurzívou

Vztahy

- Asociace

Vnější objekt získává referenci na vnitřní objekt **zvenku a dočasně** (není jeho vlastnictvím)

Objekty spolu komunikují

Graficky: čára

- Agregace, kompozice (v UML 2.0)

Druh asociace

Vnější objekt obsahuje vnitřní (ten je jeho součástí)

Graficky: čára s kosočtvercem (u vnějšího objektu), ve v2.0 vyplněný kosočtverec

- Dědičnost, generalizace-specializace

Potomek rozšiřuje předka

Graficky: šipka

10.3.2. Diagram balíčků

10.3.3. Diagram komponent

10.4. Diagramy chování

10.4.1. Diagram užití (use case diagram)

Popisuje chování systému z hlediska uživatele

Prvky

- Typ užití (kolečko)
- Aktor – uživatel (kategorie uživatelů), správce, jiný systém... (panáček)

Výhody

- Ukazuje způsoby použití aplikace a tím specifikuje požadavky
- Srozumitelnost pro zákazníka

10.4.2. Stavový diagram

Popisuje stavy, ve kterých se může objekt nacházet

10.5. Diagramy interakcí

10.5.1. Sekvenční diagram

Ukazuje objekty v čase, interakce mezi nimi (objekty si posílají zprávy)

11. Návrh uživatelského rozhraní

Skripta: ZSWI - OOpřístup kap.4

(význam, oddělené UI, charakteristika zákl. ovládacích prvků pro UI Windows)

Přívětivé uživatelské rozhraní

Hlavní typy aplikací

- Okenní (GUI, tlustý klient)
- Webové (tenký klient, ve www prohlížeči)
- Textové (na konsoli)

11.1. Oproti procedurální aplikaci

Inicializace aplikace
Obecná nastavení
Vytvoření hlavního objektu
Čekání na události a jejich obsluha
Ukončení programu

11.2. Komunikace s uživatelem

- Výběr z menu
Nástrojová lišta
Kontextové menu
Lišta s ikonami
- Dialog dotaz-odpověď
Hlavně v průvodcích
Dříve ve znakovém režimu
- Příkazová řádka
Uživatel musí znát jazyk příkazů
- Přímá manipulace
Ikony, myš, tažení objektů

11.3. Ovládací prvky

11.3.1. Vstupní

- Textové pole
- Zaškrtačací políčko (check box)
- Skupina přepínačů (radio button group)
- Seznam (list box)
- Rozbalovací seznam (drop-down list)
- Kombinovaný seznam (combo box)
- Spin box (šipky nahoru/dolu, textové pole)

11.4. Okna ve windows

SDI – Single dokument interface

MDI – Multiple dokument interface – více oken uvnitř hlavního okna

11.5. Doporučení při návrhu UI

- Podle typu uživatele
- Dodržování konzistence (posloupnost akcí, terminologie...)
- Zkrácené povely – klávesové zkratky pro pokročilé uživatele
- Odezva – na každou akci uživatele
- Předcházet chybám uživatele
- Možnost „zpět“

12. Tvorba systému nápovědy

Skripta: ZSWI - OOpřístup kap.4

(druhy uživatelské nápovědy, formáty nápovědy, nástroje pro tvorbu nápovědy)

12.1. Typy nápovědy

- Online nápověda – vyvolána přímo z aplikace
 - Tlačítko ?
 - Klávesa F1
 - Help
 - Skrytá nápověda - tooltip
- Tištěná dokumentace
- Průvodce

12.2. Formáty nápovědy

- WinHelp – starší druh nápovědy
- Microsoft HTML help – ve 32b windows
- JavaHelp – 100% JAVA
- WebHelp – přenositelný, v jakémkoli prohlížeči
- Oracle Help for Java – řešení od Oraclu

12.3. Dokumentace aplikace

12.3.1. Autorská

Programová a projektová dokumentace

12.3.2. uživatelská

13. Architektura programového systému

Přednáška: 9

Skripta: ZSWI - ZaklTémata kap.5
(typy architektur)

- Globální architektura – hrubý návrh celého IS
- Dílčí architektura
 - Funkční – hierarchie funkcí systému
 - Procesní – návrh procesů v podniku
 - Datová
 - Technologická
 - Softwarová
 - Hardwarová

13.1. Softwarová architektura

13.1.1. Lineární

Jednoduchost, průhlednost, nepřizpůsobivost ke změnám

13.1.2. hierarchická

stromová struktura

13.1.3. síťová architektura

nedefinuje pravidla pro vztah mezi částmi, neexistuje vztah nadřazenosti a podřazenosti komponent
flexibilita při přidávání nových prvků

13.1.4. vrstvená architektura

rozložení úloh, které aplikace vykonává do vrstev

13.2. technologická architektura

13.2.1. režim zpracování

- dávkové
- interaktivní
- řízené událostmi

13.2.2. rozmístění

- centralizované zpracování – hlavní počítač mainframe a terminály
- decentralizované – samostatné off-line počítače, vzniká nekonzistence dat
- distribuované zpracování – několik serverů

13.2.3. monolitické aplikace

v současnosti malé a vysoce specializované aplikace

13.2.4. modulární aplikace

rozdělení problému na menší části, komponenty

13.2.5. architektura klient/server

- prezentační vrstva
- aplikační vrstva
- datová vrstva

14. Životní cyklus programového systému *

Přednášky: 8,10

Skripta: ZSWI - ZaklTémata kap.5

14.1. Úvodní studie (UST)-

studie proveditelnosti,

je zaměřena na detailní posouzení realizovatelnosti požadavků na projekt
provádí se variantní návrh koncepce řešení projektu

14.2. Příprava plánu projektu

určení vedoucího projektu,

určení členů týmu,

definování termínů definování ekonomických charakteristik

Závěrem je rozhodnutí, zda je projekt za daných požadavků, dostupných technologií, zdrojů a rozpočtu možné realizovat

14.3. Specifikace požadavků

dokument Specifikace požadavků

14.4. Globální analýza a návrh (GAN)

vymezení hlavních funkcí a dat projektovaného systému na konceptuální úrovni.

Use case diagram

konceptuální Class diagram

14.5. Detailní analýza a návrh (DAN)

transformuje konceptuální úroveň návrhu do technologické, která je závislá na zvoleném implementačním a provozním prostředí aplikace.

Class diagram na technologické úrovni

sekvenční diagramy

14.6. Fáze návrhu

bývá někdy rozdělována na dvě fáze:

systémový návrh,

objektový návrh.

V rámci systémového návrhu je třeba zvolit implementační prostředí a architekturu aplikace. Fáze objektového návrhu definuje třídy a vztahy mezi třídami, algoritmy metod a uživatelské rozhraní.

14.7. Implementace (IM)

je transformace technologické úrovně návrhu IS do implementační úrovně,

14.8. Testování

14.9. Zavádění (ZA)

instaluje se systém,

transformuje se původní datová základna do stavu potřebného pro novou verzi IS/ICT,

školí se uživatelé aplikace

realizuje se zkušební provoz aplikace. Úspěšné završení fáze zavádění končí akceptací a uzavřením projektu vývoje.

14.10. Provoz a údržba (PU)

aplikace je provozována

provádí se údržba – realizuje se změnové řízení – zaznamenávají se požadavky na změny a po schválení se realizují

15. Metodiky vývoje programového systému

Přednáška: 8

Skripta: ZSWI - ZaklTémata kap.6

15.1. rigorózní metodiky

dnes často nevyhovují
vyžadují požadavky specifikované předem
v průběhu se už těžko dělají změny

15.2. agilní metodiky

nepopisují procesy, ale principy
zaměřeny na činnosti, které vytvářejí hodnotu

- individualitám a komunikaci před procesy a nástroji
- provozuschopnému software před obsažnou dokumentací
- spolupráci se zákazníkem před sjednáváním kontraktu
- reakci na změnu před plněním plánu

příklady agilních metodik

- Adaptive Software Development (ASD),
- Dynamic Systems Development Method (DSDM),
- Feature-Driven Development (FDD),
- Extreme Programming (XP),
- Lean Development,
- SCRUM,
- Crystal metodiky

Charakteristika

- iterativní vývoj s velmi krátkými iteracemi,
- zaměření na fungující SW, který má hodnotu pro zákazníka,
- lidé jsou prvořadým faktorem – důraz na spolupráci a komunikaci,
- tolerantní ke změnám,
- automatizované testování

16. Webové aplikace a jejich technologie *

Přednáška: 9

(Charakteristika webových aplikací a technologií pro jejich tvorbu)

Webová aplikace

je založena na C/S architektuře - tenký klient

funkcionalita aplikace probíhá na více počítačích v rámci internetu

klient - prohlížeč (browser)

server - webový server

příklady elektronický obchod, internetové aukce, předpověď počasí pro danou lokalitu, apod.

Prohlížeč - browser

představuje uživatelské rozhraní webové aplikace

umožňuje zobrazovat WWW stránky a navigovat mezi stránkami

HTML - HyperText Markup Language

standardizovaný jazyk pro vytváření WWW stránek

značkovací jazyk

HTTP

Protokol pro komunikaci mezi klientem (prohlížeč) a serverem (web)

Používá TCP/IP protokol (většinou port 80)

Neudrhuje se spojení (tzv. connectionless),

Data nejsou nijak šifrována a není problém je odposlechnout, případně zneužít
HTTPS - digitální certifikáty

Ověřuje identitu serveru nebo i klienta

Šifrování komunikace – není možné odposlouchávat

Vydávají certifikační autority

Statické stránky

internetová revoluce,

skvělé pro sdílení informací

omezení:

Jak vytvořit aplikace ?

Jak zobrazovat existující data ?

Dynamicky generované stránky /zpracování na straně serveru)

Prohlížeč(klient)>Žádost>Web(server) >Zpracování> Odpověď

Paralelní trend- Kód na straně klienta- zpracovává klient

Jednoduché akce – kontrola vstupů, obohacení uživatelského rozhraní

Možné jazyky:VBScript, Javascript ,Java aplety, ActiveX

Serverové zpracování – PHP, ASP, JSP, CGI

používání skriptů-do HTML textu jsou vepsány části kódu, server

zpracuje kód, zbytek ponechá beze změny

pohodlný vývoj, zlepšený výkon, dobrá škálovatelnost

17. Typologie vývojových nástrojů *

Přednáška: 10

(trendy, kritéria pro výběr)

komplex programových prostředků pro design, vytvoření, testování a dokumentaci programových systémů

Nástroje pro vývojáře

-Správa požadavků

-Konfigurační řízení- řízení verzí , týmová spolupráce

-Řízení změn-Change management

SW se neustále mění – důvody

iterativní vývoj

chyby a nutnost jejich odstranění

požadavky na změny ze strany uživatelů

změny vyvolané změnami technologií

-Testovací nástroje- jeden z hlavních způsobů zajištění kvality SW

automatizované testy, které prověří všechny prvky, zátěžové testy

-Instalační nástroje

Správa požadavků

Vizuální modelování,

generování kódu

Konfigurační management

Automatizované testy

Dokumentace

Řízení projektu

Vývojová prostředí

liší se podporou komponentových standardů – COM, CORBA, EJB
dnes většina podporuje webové služby
doplnění stávajících nástrojů– např. Delphi 6 a vyšší
nástroje speciálně zaměřené na WS
– Visual studio .NET

18. Principy obj. přístupu, základní vlastnosti objektů

Přednášky: 1, 2, 3

Skripta: ZSWI - OOpřístup kap.2, ZSWI - ZaklTémata kap.4,5

19. Charakteristika OO programovacích jazyků

Přednášky: 1, 2, 3

Skripta: ZSWI - OOpřístup kap.2, ZSWI - ZaklTémata kap.4,5

20. Třídy a objekty, vztahy tříd, diagram tříd UML

Přednášky: 1, 5

Skripta: ZSWI OOpřístup kap.3

21. Generalizace - specializace

Přednášky: 1,2,3, 5

Skripta: ZSWI - OOpřístup kap.3

(Vztah generalizace-specializace mezi třídami a možnosti jeho implementace, ukázka na příkladě zvoleného programovacího jazyka)

22. Dědičnost

Přednášky: 1,2,3, 5

Skripta: ZSWI - OOpřístup kap.2, 3

(výhody a nevýhody použití, předek všech tříd)

23. Polymorfismus

Přednášky: 1,2,3, 5

Skripta: ZSWI - OOpřístup kap.2, 3

(princip, jeho realizace, ukázka na příkladě zvoleného programovacího jazyka)

- různé objekty mohou reagovat na stejnou zprávu různě
- vlastnost, která umožňuje pojmenovat určitou akci jedním jménem, přičemž její implementace se v jednotlivých třídách může lišit

24. Vytváření a rušení objektů

Přednášky: 1,2,3

Skripta: ZSWI - OOpřístup kap.2, 3

(ukázka na příkladě zvoleného programovacího jazyka)

25. Seznamy objektů

Cvičení

Skripta: ZSWI - OOpřístup kap.5, 6

(princip, příklad ve zvoleném programovacím jazyce)

- Seznam - List (ArrayList)

Lineární struktura

Přístup: sekvenčně, nebo podle pozice

- Mapa (map)

Prvky jsou přístupné pomocí klíče

- Množina (set)

Kontejner, který dokáže odpovídat na dotaz, jestli obsahuje objekt X

26. Komponenta

Přednášky: 11

Skripta: ZSWI - ZakITémata kap.5.5

(komponenta, interface, vztah komponent a tříd)

27. Komponentový vývoj

Přednášky: 11

Skripta: ZSWI - ZakITémata kap.5.5

(základní infrastruktury komponentní technologie, příklady)