

> **co to je zapouzdreni**

- sjednocení atributu a metod do jednoho celku
- vymezuje kompetence a poslání objektu
- zabezpečuje před vnějším zásahem

> **co je zpráva**

- mechanismus implementován jako volání metody
- způsob jak se dostat přes zapouzdření
- poslaný příkaz jednoho objektu druhému

> **co je to trída**

- více objektu se stejnými atributy a metodami

> **co je to UML**

- vizuální modelovací jazyk, který slouží je specifikaci, tvorbě a dokumentaci softwarových systémů

> **k čemu slouží konstruktor**

- musí se zavolat před použitím objektu
- slouží pro inicializaci
- naplní ještě před svý vykonáním VMT

> **k čemu slouží destruktorka**

- zruší tabulku VMT
- volá se na konci práce s objektem

> **jak se zakresluje třídy v UML**

- obdelnik, název tučně s velkým písmenem na začátku

> **vyhody trojvrstevního návrhu aplikace**

- vrstvu je možno vyměnit za jinou, může pracovat více programátorů na jednom projektu, program je přehlednější (je dělen do 3 částí)

> **jake třídy jsou kompatibilní**

- potomek a předek

> **rozdíl mezi třídou a objektem**

- třída je vyšší abstrakcí objektu

> **co je to objekt**

- je to samostatná datová struktura tvořená podle axiomu
- má stav a chování

> **rozdíl mezi abstrakcí a asociací**

- asociace - objekt může odkazovat na vnější objekt - oba mohou existovat nezávisle na sobě
- agregace - objekt může mít jiný objekt jako svou autonomní část - vnější objekt bez vnitřního ztrácí význam

> **časová vazba**

- volá už při překladu
- slouží k spárování volání metod s jejich umístěním v paměti

> **pozdí vazba**

- volá se až za běhu programu
- slouží k zjištění umístění volané metody v paměti

> **co je to polymorfismus**

- vlastnost, umožňuje použití jednoho stejného jména pro více souvisejících ale technicky různých účelů

> **co je to virtuální metoda**

- metoda která je deklarovaná ale není určeno co vykonává

Objekt: Má stav a chování

Stav je uložen v proměnných – attributech

Chování implementují funkce – metody

Žádný atribut ani metoda nejsou vidět z vnějšku -> tato vlastnost se nazývá zapouzdření – encapsulation

Zpáva: Jak se dostat přes „slupku“ zapouzdření?

Skřez slupku objektu vytvořím vstupně/výstupní kanál, přes který pošlu objektu zprávu a přijmu odpověď na ni.

Tento mechanismus je ve většině OO jazyků implementován jako volání metody.

Třída: Více objektů má stejné vzorce chování (metody) a stejné atributy

Alík, Punťa i Rex jsou Psi

mají stejné metody – umí štěkat, běhat, žrát, spát,...

mají stejné atributy – jméno, srst, oči, výšku,...

Atribut(proměnná):

Metoda(funkce, procedura):

Zapouzdření - Veškeré atributy a metody jsou viditelné v celém modulu,

- Mimo modul (tj. je-li třída definována v samostatném modulu)
- se pro řízení přístupu používají klíčová slova public a private

Dědičnost/zobecnění: Potomek dědí od předka všechny atributy a metody.

Dědičnost: Třída zaměstnanec dědí atributy a metody od třídy člověk. Zaměstnanec je člověk.

Předek, Potomek: Třída člověk je předkem třídy zaměstnanec. Třída zaměstnanec je potomkem třídy člověk.

Zaměstnanec je člověk.

Abstraktní metoda: Metoda, která nemá žádnou implementaci.

Polymorfismus: instance různých tříd na stejný podnět (na vyvolání stejné metody) reagují různě.

Instance více tříd poskytují svému okolí stejnou službu, ale každá instance na vyžádání této služby provede něco jiného.

Virtuální metody: Pro použití virtuálních metod musí být splněny následující podmínky:

- Virtuální metoda musí mít stejnou deklaraci (tj. stejné atributy a návratová hodnota).
- Třída musí obsahovat speciální metodu nazývanou konstruktor Napr.: constructor init;
- Konstruktor ještě před vykonáním svého kódu naplní VMT.
- Konstruktor musí být volán před prvním použitím objektu.
- Použijete-li virtuální metodu před voláním konstrukturu, jsou následky nepředikovatelné (nejčastěji se program zhroutí i s DOSem).

constructor: je speciální metoda, která inicializuje objekt obsahující virtuální metody

destructor: je speciální metoda, která uvolní dynamický objekt z haldy

virtual: metoda jejíž volání je spojeno s kódem až při běhu programu. Pozdní vazba.

New: vloží novou dynamickou proměnnou a vloží její adresu do proměnné typu ukazatel

Tabulka virtuálních metod(VMT – Virtual Methods Table): Tabulka obsahuje ukazatele na metody které se mají zavolat, pokud se program chystá provést některou z virtuálních metod.
- I Každý objekt si nese VMT s sebou a má ji unikátní stejne jako atributy.

Včasná vazba: již na začátku (při překladu) jsou spárovány volání metod s jejich umístěním v paměti.

Pozdní vazba: adresa metody se určuje až při volání.
- Budeme používat při použití některých predefinovaných metod.
- Tento způsob nakládání s metodou je třeba explicitně přikázat.

Asociace: Objekt může obsahovat odkaz (ukazatel) na vnější objekt. Např. Auto-řidič.
Oba objekty mohou existovat (a mají smysl) i jeden bez druhého.

Agregace: Objekt může mít jiný objekt jako svou autonomní část. Např. Auto-motor.
Oba objekty mohou existovat i jeden bez druhého. Vnější objekt (auto) bez vnitřního (motor) ztrácí funkčnost.

UML(Unified Modelling Language): je jazyk, který slouží ke specifikaci, vizuálnímu popisu, tvorbě a dokumentaci jednotlivých součástí softwarových systémů.
Slouží k přehlednému návrhu modelu systému od celkového přehledu.

Dědičnost - šipka s prázdným koncem. Nevyplněná uzavřená šipka. --->

Asociace - normální šipka --->

Agregace - nevyplněná ---◇

Kompozice - vyplněná ----◇

Trojvrstvý model: Dobře napsaný program (informační systém) se skládá z několika logicky oddělených částí:

- Vizualizace údajů uživateli (UI, WEB, tenký klient, . . .),
 - ukládání a správa dat (databáze, filesystém, . . .),
 - zpracování dat.
- Aby byl program dobře spravovatelný, je vhodné jednotlivé logické části oddělit do nezávislých částí – bloku, vrstev. Jednotlivé vrstvy spolu komunikují pomocí rozhraní.

private : atributy - privátní (neverejné) paměť objektu, která uchovává aktuální stav objektu.
metody - privátní (neverejné) operace nad atributy, které mění stav objektu.

public : deklarované metody (atributy zde asi nebudou) lze vyvolat z jiného modulu

Dispose: Vyvolá destruktory a uvolní objekt z haldy. Rozšířené pro práci s objekty. Dispose(p:Pointer, Done:Destructor);

Fail: - Lze ji použít pouze v konstrukturu. Značí, že v průběhu vykonávání konstrukturu došlo k chybě a konstruktor se má přerušit.
- Způsobí, že konstruktor dealokuje dynamickou instanci objektu na haldě.
- Do ukazatele na objekt se přiřadí hodnota nil a objekt je nepoužitelný.

inherited: znamená zdědění Klíčovým slovem Inherited vyvoláte metodu předka
Použití u zděděných metod, konstrukturu i destrukturu.