



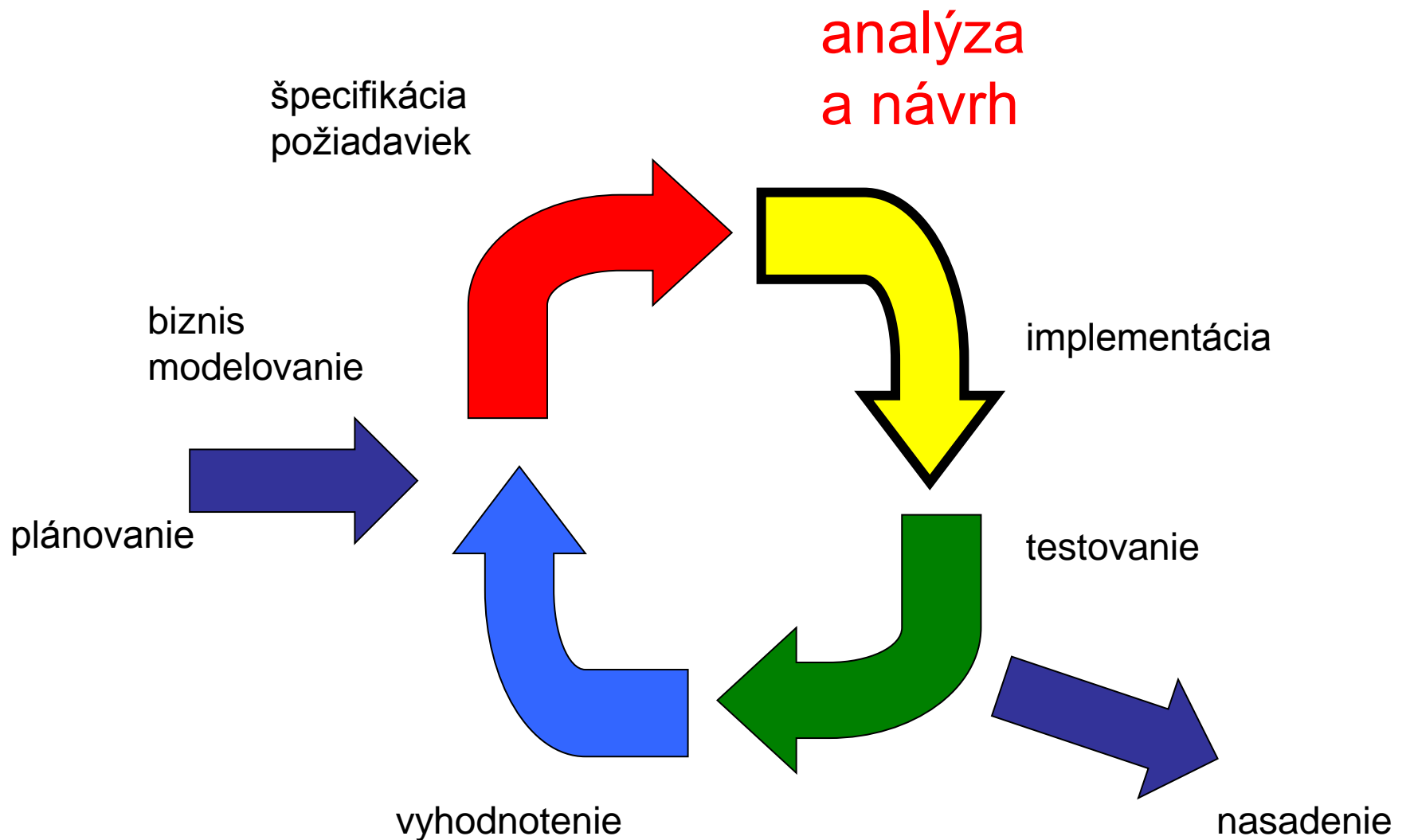
Analýza

- Čo si predstaviť pod analýzou?
- Analytická trieda
- Metódy identifikácie tried
- Analýza balíčkov

RUP



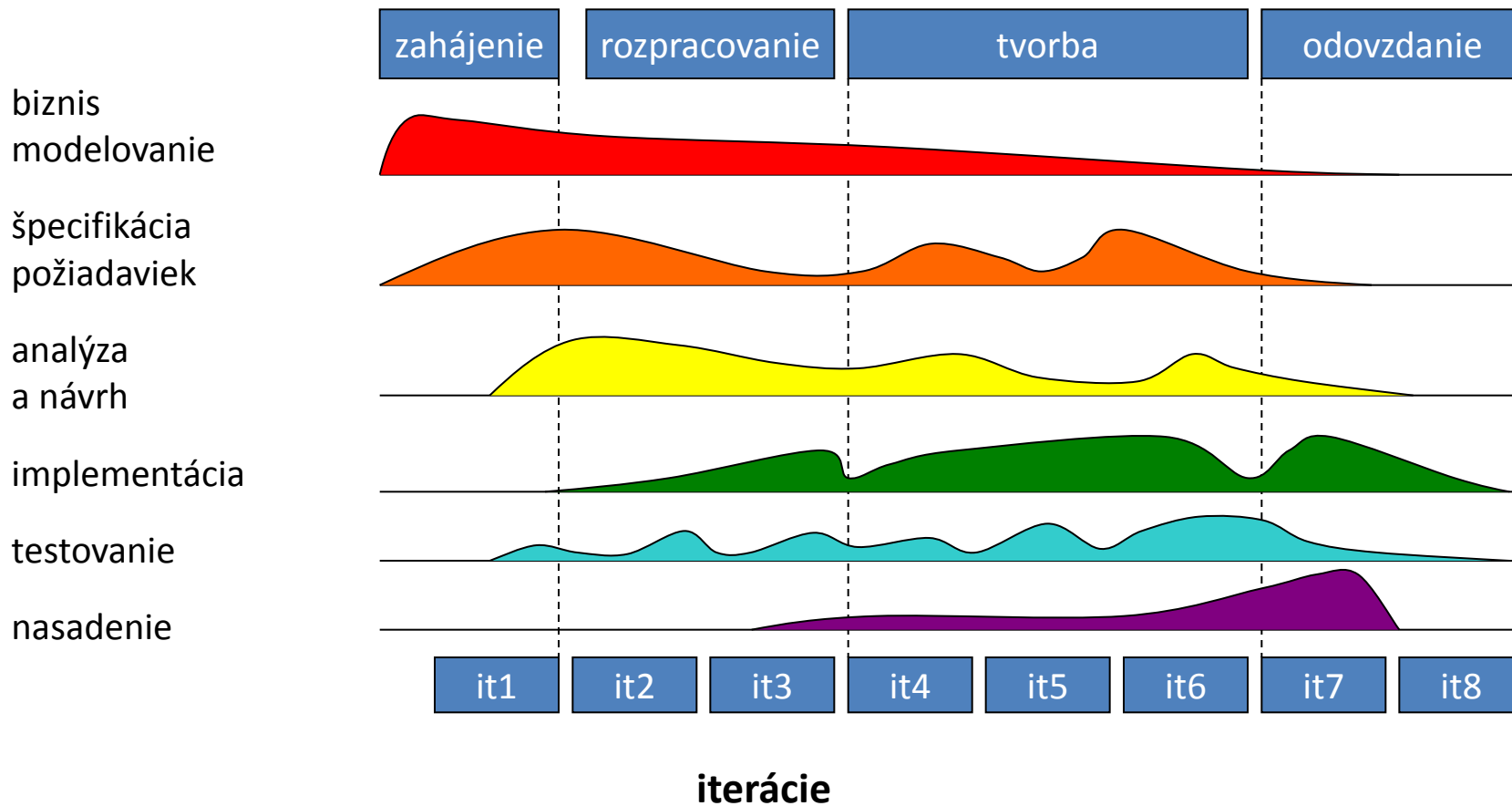
Analýza a návrh



RUP – schéma (obsah x čas)

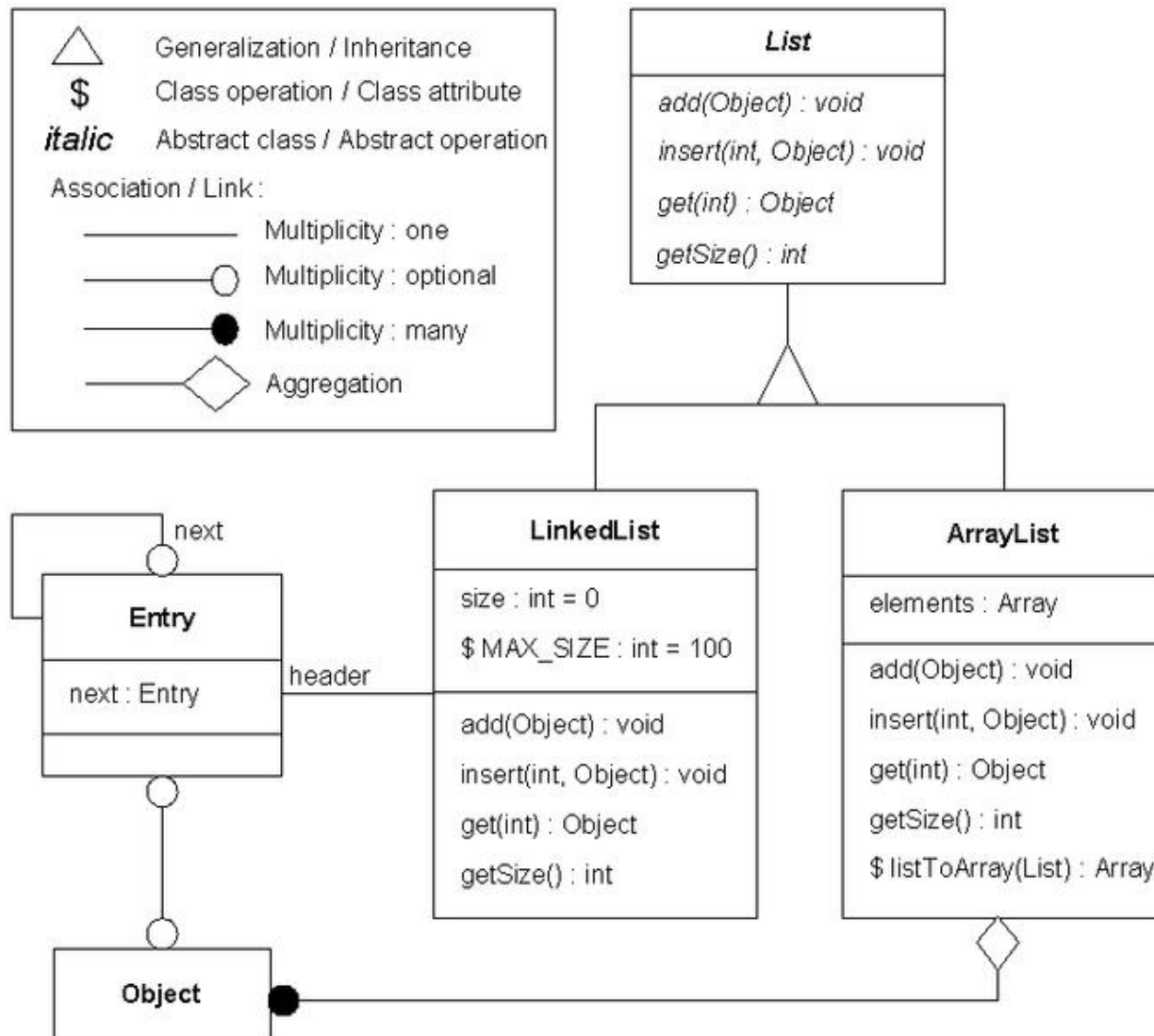
tok činností

fázy



Objektovo – orientované metódy

- 1984 -1994 – návrh viac než 40 nových objektových metodík
 - OMT - Rumbaugh et al. 1991
 - OOSE - Jacobson 1992
 - OOD - Booch 1994
- Integrácia metodík autorov Rumbaugh, Jacobson a Booch – vytvorenie metodiky RUP - Rational Unified Process





Proces rozdelenia komplexného problému na menšie časti, za účelom ich lepšieho pochopenia

- Cieľ:
 - Vytvoriť analytický model – konceptuálny model
- Zachytenie podstatných požiadaviek a charakteristických rysov systému

Na aké menšie časti?



Objektovo orientovaná analýza a návrh

- Prístup modelujúci systém ako skupinu spolupracujúcich objektov

Objektovo orientovaná analýza

- Aplikuje techniky objektového modelovania za účelom analýzy funkcionálnych požiadaviek

Objektovo orientovaný návrh

- Spracováva analytické modely za účelom vytvorenia špecifikácie implementácie.



Aké sú základné princípy objektového programovania?



Rozdelenie systému na vrstvy

Prezentačná vrstva

Logická vrstva

Dátová vrstva

Analytický model

Diagram
tried

Diagram
spolupráce

Diagram
nasadenia

Sekvenčný
diagram

Stavový
diagram

Diagram
balíčkov

Pravidlá tvorby diagramov

- Tvorený v doménovom jazyku
- „Rozprávajte príbeh“
- Perspektíva
- Rozlišujte problematiku domény a riešenia
- Minimalizácia vzťahov
- Len „prirodzená dedičnosť“
- Tvoríť model pre maximálny počet používateľov
- Čo najjednoduchší

- Architektonická analýza
- Analýza tried
- Analýza balíčkov
- Analýza prípadov použitia

Analýza

Vstupy

- Doménový model
- Model požiadaviek
- Model prípadov použitia
- Popis architektúry

Výstupy

- Architektonická analýza
- **Analytické triedy**
- Analýza balíčkov
- Realizácie prípadov použitia

Analytická trieda



Analytická trieda – trieda, ktorá reprezentuje základné dáta a chovanie, a ktorá nezachytáva softvérové a hardvérové podrobnosti

- Názov odráža jej účel
- Hrubá abstrakcia, špecifický prvok domény
- Mapuje jasne identifikovanú vlastnosť
- Mala množina zodpovedností
- Súdržná
- Minimum väzieb

Praktické rady

- 3-5 zodpovedností
- Žiadna trieda nie je sama o sebe
- Nie veľa malých tried
- Nie niekoľko veľkých tried
- Nie „funkcoidy“
- Nie všemocné triedy
- Nie hlboké hierarchie dedičnosti

Identifikácia tried

- Analýza podstatných mien a slovies
- Metóda CRC
- Metóda stereotypov RUP
- Z iných zdrojov



Analýza podstatných mien a slovies

- Podstatné mená a frázy – triedy, atribúty
- Slovesá a slovesné frázy – zodpovednosti, operácie tried
- Pozor na:
 - Nežiaduce triedy
 - Nepresné pochopenie domény
 - Skryté triedy



Metóda CRC

- Class, responsibilities & collaborators
- Spolu s metódou analýzy podstatných mien a slovíes
- Oddelenie zhromažďovania informácií a ich analýzy



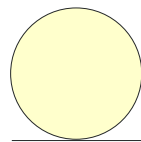
WORD DICTIONARY	
Responsibilities	Collaboration
Store & recall words from an external file (easy data-file/ hard data-file)	Created by games 1-3 to return random words
Store & recall sentences	Accesses data files to return random words/ sentences

- Zhromažďovanie informácií
 - Prijímanie všetkých nápadov a ich zaznačenie
 - Pomenovanie „predmetov“ domény
 - Kandidáti na triedu alebo atribút
 - Uvedenie zodpovednosti predmetov
 - Práca v tíme
 - Označenie tried, ktoré by spolupracovali
- Analýza informácií

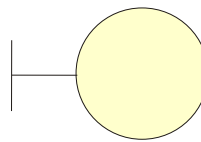
Metóda stereotypov RUP



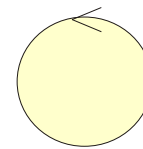
- Triedy je možné rozdeliť do troch stereotypov:
 - Hraničné triedy – označenie <<boundary>>
 - všetko s čím priamo komunikujú aktéri
 - Napr. formuláre, komunikačné protokoly, rozhrania
 - Entitné triedy - označenie <<entity>>
 - Obsahujú informácie, ktoré systém udržiava dlhšiu dobu, zodpovedajú objektom z reálneho sveta
 - Napr. študent, fakulta, predmet, a pod.
 - Riadiace triedy - označenie <<control>>
 - Koordinujú správanie sa systému
 - Napr. Triedy obsahujúce riadiacu logiku, nastavujúce obsah entitných tried



Order



OrderForm



OrderControl

Prvá verzia

- Porovnať zdroje informácií
- Zlúčte analytické triedy, atribúty a zodpovednosti
- Zadaťte ich do CASE nástroja
- Spolupracovníci reprezentujú vzťahy
- Vylepšite pomenovanie tried a atribútov

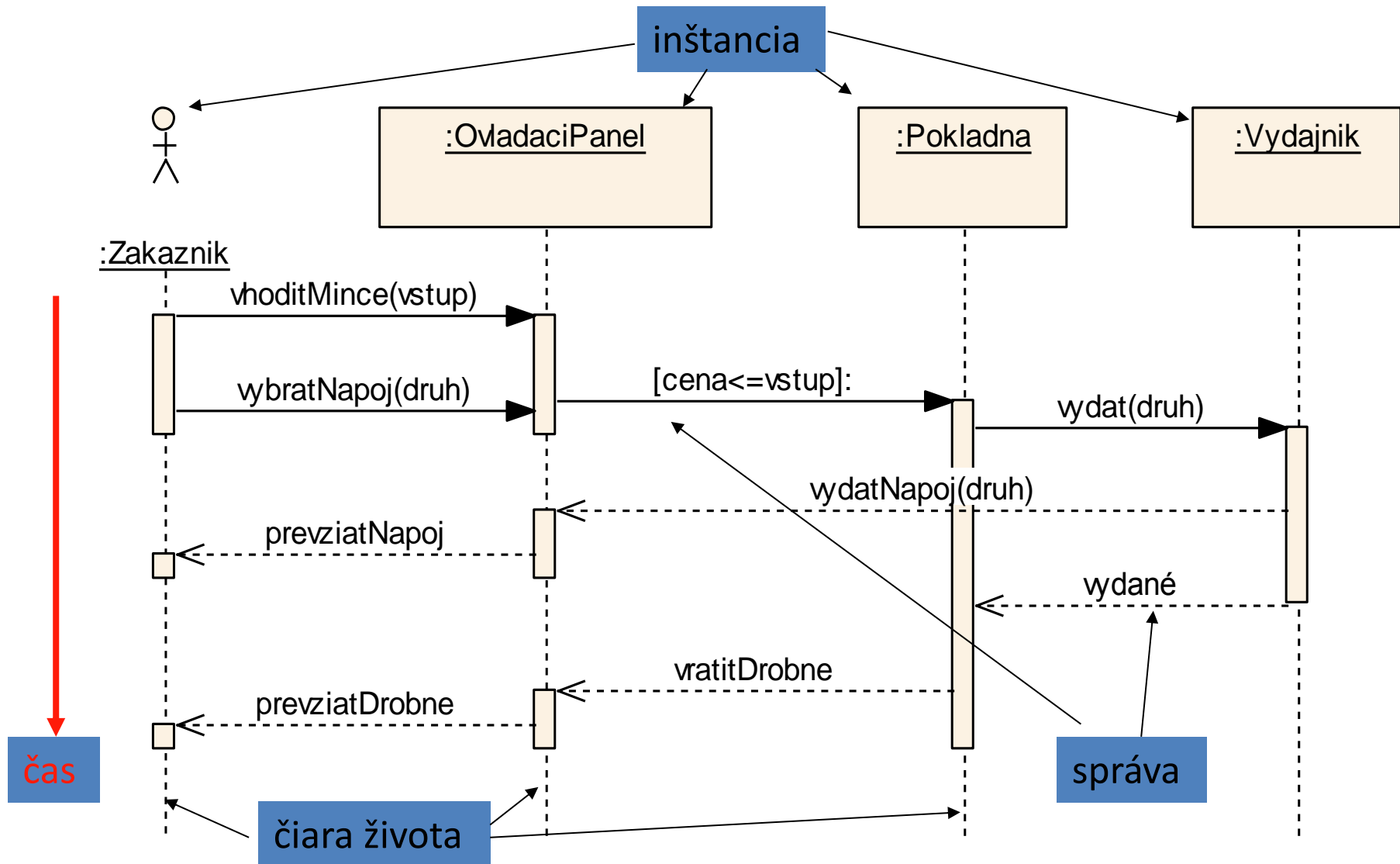
Sekvenčný diagram



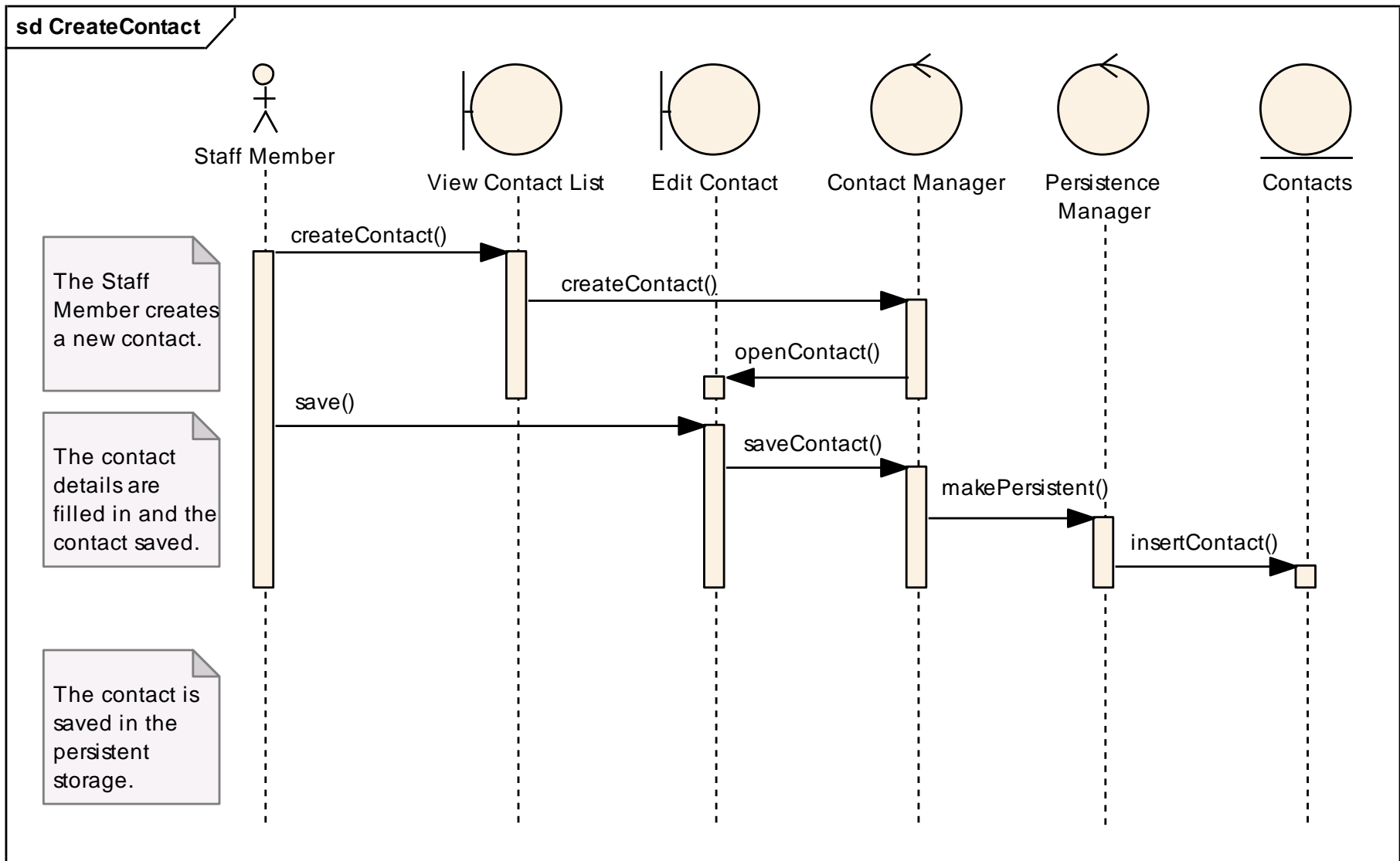
Zobrazuje časovo utriedenú interakciu medzi objektmi za účelom vykonania podstatných častí prípadu použitia

- Vychádza zo scenára
- Postup
 - Identifikujte aktorov
 - Špecifikujte hraničné objekty a jeden riadiaci objekt
 - Odvodte entitné objekty
 - Zadefinujte správy medzi nimi

Sekvenčný diagram



Sekvenčný diagram



- Aktori by mali komunikovať iba s hraničnými objektmi
- Hraničné objekty len s riadiacimi a aktormi
- Riadiace objekty so všetkými, okrem aktorov

Pravidlá

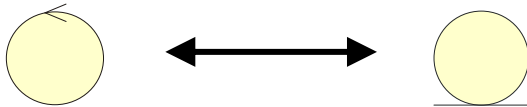
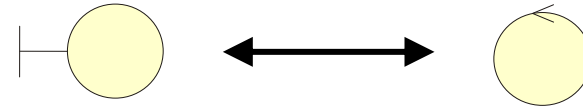
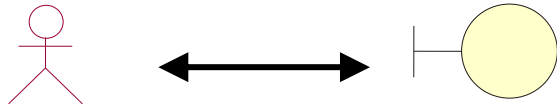
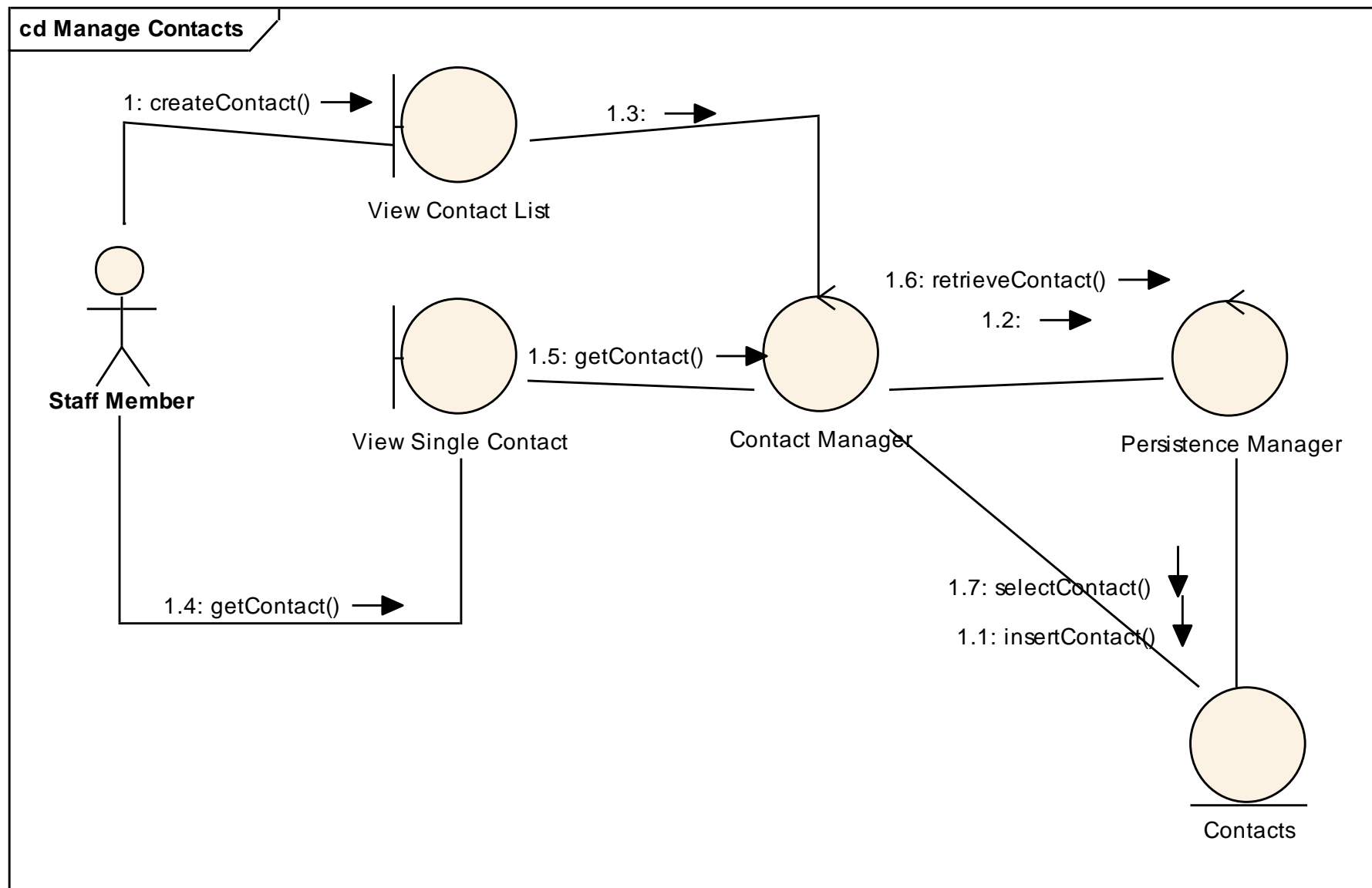


Diagram spolupráce

- Zobrazenie podobnej informácie ako v sekvenčnom diagrame
- Nezobrazuje čas
- Zobrazuje vzájomnú spoluprácu objektov

Diagram spolupráce



Analýza

Vstupy

- Doménový model
- Model požiadaviek
- Model prípadov použitia
- Popis architektúry

Výstupy

- Architektonická analýza
- Analytické triedy
- **Analýza balíčkov**
- Realizácie prípadov použitia

Analýza balíčkov

- Zoskupovanie tried



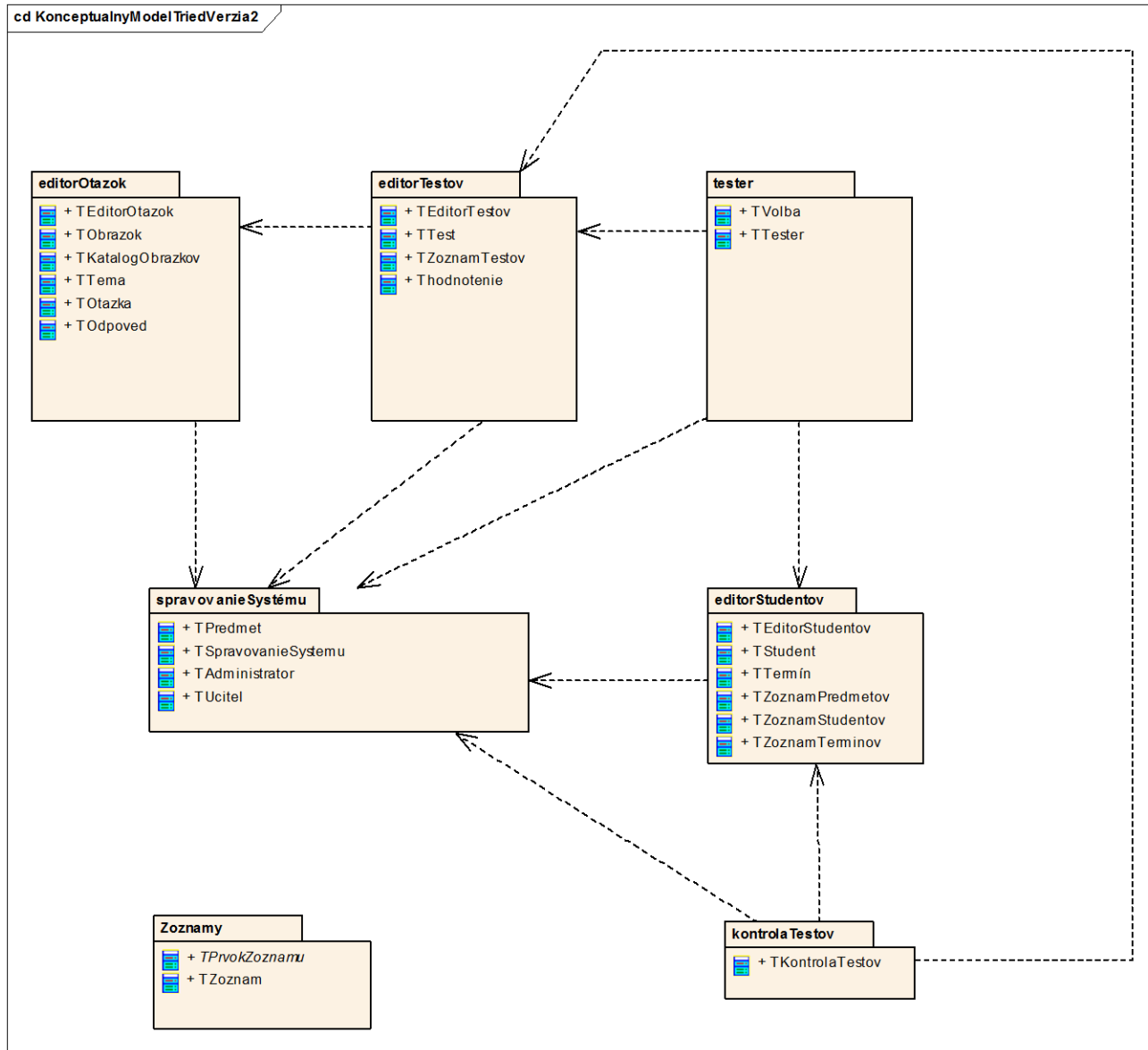
Abstrakcia združovania – je to kontajner a vlastní modelovaných prvkov

- Vlastný menný priestor
- Univerzálny mechanizmus zoskupovania prvkov a diagramov

Umožňujú

- Súbežnú prácu
- Zoskupovanie sémanticky súvisiacich prvkov
- Definovanie hraníc vo vnútri modelu
- Zapuzdrený menný priestor
- Vnáranie balíčkov

Package diagram



Identifikácia balíčkov

- Skupiny prvkov silne sémanticky previazané
- Súdržné skupiny tried
- Hierarchie dedičnosti
- Môže pomôcť aj rozdelenie medzi prípadmi použitia
- Minimalizácia vzťahov
- Čo najjednoduchší model
- Zabrániť cyklickým závislostiam

Architektonická analýza

- Zoskupovanie tried do množiny súdržných balíčkov
- Štruktúrované do oddielov a vrstiev
- Minimalizácia vzťahov – hlavný cieľ
 - Minimalizácia závislostí medzi balíčkami
 - Minimalizácia verejných prvkov
 - Maximalizácia súkromných členov
- Špecifické a univerzálne vrstvy

Analýza

Vstupy


- Doménový model
- Model požiadaviek
- Model prípadov použitia
- Popis architektúry

Výstupy

- Architektonická analýza
- Analytické triedy
- Analýza balíčkov
- **Realizácie prípadov použitia**

Realizácia prípadov použitia

- Modelované interakcie medzi objektmi

 Popis spolupráce inštancií analytických tried za účelom dosiahnutia požadovaného chovania systému

- Ciele:
 - Zistenie interakcií analytických tried
 - Zisťovanie zasielaných správ
 - Kľúčové operácie
 - Kľúčové atribúty
 - Kľúčové relácie
 - Aktualizácia modelov

Zloženie

Prvok	Popis
Diagramy analytických tried	Interakciu analytických tried
Diagramy interakcie -komunikačné diagramy -sekvenčné diagramy	Spolupráca a interakcia špecifických inštancií
Špeciálne požiadavky	Odhalenie nových požiadaviek
Spresnenie prípadov použitia	Aktualizácia prípadov použitia

- Definícia problému
 - Vytvorte softvér pre sieť bankomatov „Našej Banky“.
 - Bankomaty budú komunikovať s centrálnym počítačom banky, ktorý transakcie autorizuje a uskutoční zmeny na účte.
 - Softvér centrálného počítača dodá banka.
 - Systém vyžaduje uchovávanie záznamov o činnosti a zabezpečení.

Príklad

- Uskutočníme doménovú analýzu s cieľom maximálneho porozumenia doméne aplikácie
- Navrhujeme základné doménové triedy, t.j. triedy reprezentujúce objekty relevantné v aplikačnej doméne je potrebné mať v aplikačnej doméne
 - Sledujú sa podstatné mená v definícii problému, veci a miesta v aplikačnej doméne, pre všetky vytvoríme predbežnú triedu
 - Eliminácia chybných a nepotrebných tried
 - Nerelevantné triedy – zrušenie
 - Ak trieda popisuje jediný nesamostatný objekt –atribút
 - Trieda popisuje činnosť objektu - operácia

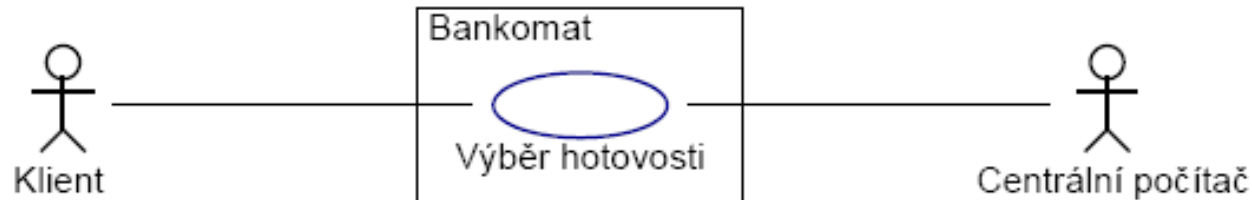
Príklad

- Predbežné triedy vyplývajúce z definície problému:
 - softvér, bankomat, centrálny počítač, banka, transakcia, účet, záznam o činnosti, zabezpečenie
- Predbežné triedy vyplývajúce z aplikačnej domény:
 - klient, platobná karta, potvrdenka, výplata
- Eliminácia vágnych tried:
 - softvér, zabezpečenie

Príklad

- Realizácia zberu požiadaviek
 - Určenie aktérov
 - Primárni a sekundárni užívatelia, externé HW a SW systémy
 - Identifikácia prípadov použitia
 - Stručný popis účelu prípadov použitia
 - Rozloženie prípadov použitia na kroky
 - Základná postupnosť aktivít a alternatívna postupnosť aktivít
 - Prípady použitia a aktérov sa štruktúrujú pomocou extend a include
 - Ak je veľa prípadov použitia, zoskupujú sa do balíčkov
- Prípady použitia, ktoré je možné ľahko prehliadnuť
 - Štart a ukončenie systému
 - Administrácia systému, a pod.

Príklad



- Prípád použitia:
 - Výber peňazí z bankomatu
- Aktéri:
 - Klient,
 - Centrálny počítač
- Stručný popis:
 - Zákazník vloží kartu a požiadá o výber určitej čiastky.
 - Bankomat mu po potvrdení centrálnym počítačom požadovanú čiastku vydá.

- Popis jednotlivých krokov základného scenára
 - Klient vloží kartu. Bankomat kartu prečíta a zistí jej sériové číslo
 - Bankomat požiada užívateľa o zadanie PIN, užívateľ zadá 1234
 - Bankomat overí číslo karty a PIN v centrálnom počítači
 - Bankomat požiada o zadanie veľkosti čiastky, užívateľ zadá 1000 Sk
 - Bankomat požiada centrálny počítač o uskutočnenie transakcie, centrálny počítač transakciu uskutoční a vráti nový zostatok účtu
 - Bankomat vydá čiastku, vytlačí potvrdenku a vráti kartu.

Príklad

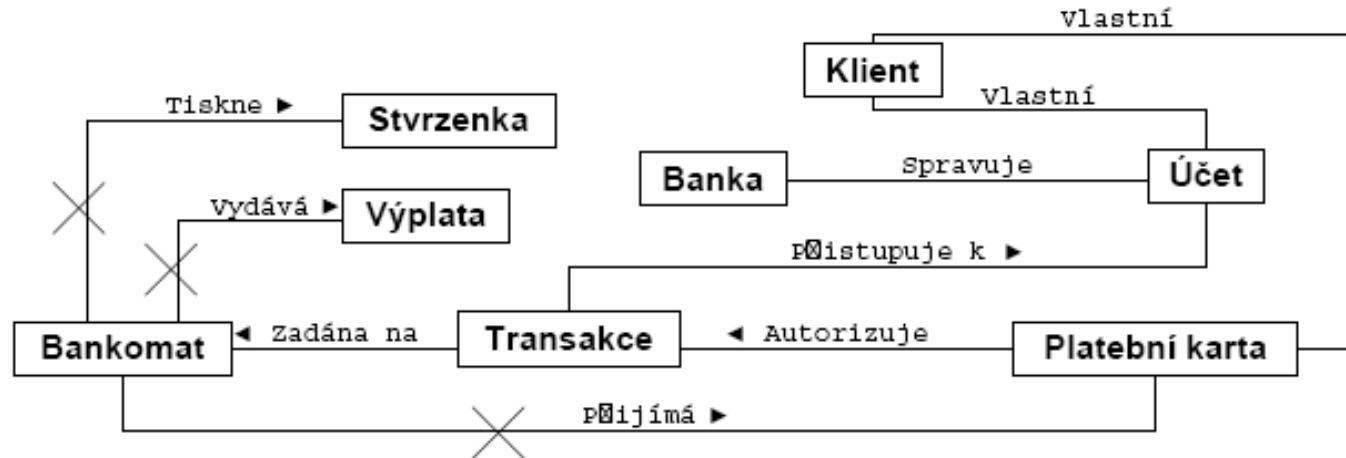
- Entitné objekty v príklade bankomatu
 - klient a účet
- Výber peňazí je overený centrálny počítačom – aktér
- Komunikácia s týmto aktérom prebieha v ATM sieti → rozhranie ATM – hraničný objekt
- Ďalšie hraničné objekty
 - klávesnica,
 - obrazovka,
 - čítačka platobných kariet,
 - výdajný automat bankomatu,
 - tlačiareň potvrdení a pod.
- Tvorba prototypu užívateľského rozhrania pre hraničné objekty
- Prototyp by mal byť abstraktný, nie veľmi konkrétny



- Tvorba predbežných asociácií medzi triedami
 - Asociácia – dlhší vzťah medzi inštanciami
 - Asociácie môžu zodpovedať fyzickému umiestneniu, vlastníckemu, riadiacemu alebo komunikačnému vzťahu
 - Asociácie by mali pomenované účelom asociácie
 - Zatiaľ nerozlišujeme medzi asociáciami, agregáciami, neuvádza sa násobnosť ani priechodnosť

Príklad

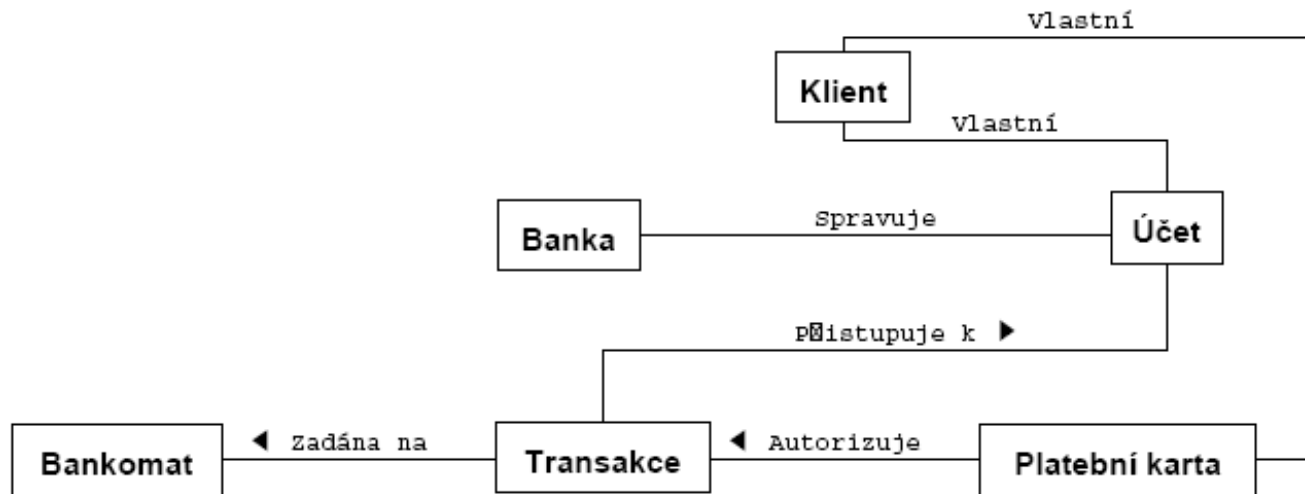
- Predbežné asociácie medzi triedami



- Optimalizácia asociácií:
 - Zrušenie nepodstatných asociácií a asociácií ktoré predstavujú popis implementácie
 - Zrušenie predbežných asociácií – jednorazové akcie (bankomat prijíma platobnú kartu, ale medzi nimi nie je trvalý vzťah)
 - Vyhýbanie sa asociáciám medzi dvoma riadiacimi objektami a medzi hraničným a riadiacim objektom
 - Vyhýbanie sa ternárnym a viacnásobným asociáciám – je možné ich preštruktúrovať na binárne asociácie

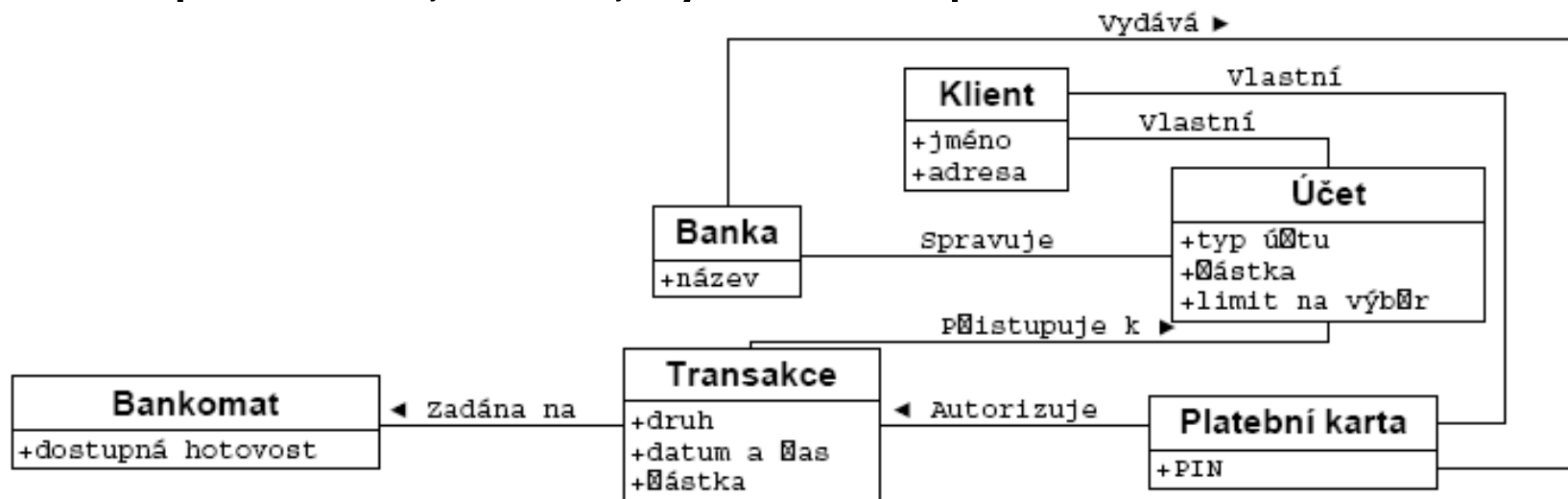
Príklad

- Tvorba agregácií a kompozícií
 - Agregácia – ak je objekt súčasťou alebo je podriadený inému objektu
 - Kompozícia – ak jeden objekt vlastní iný objekt a majú rovnakú dobu života



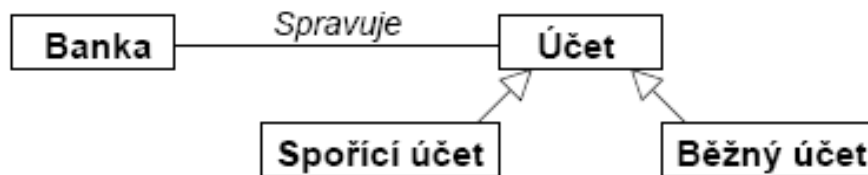
Príklad

- Hľadanie primárnych atribútov objektov a asociácií
 - Hľadáme najdôležitejšie logické atribúty, ktoré sú relevantné pre aplikáciu
 - Navonok viditeľné vlastnosti jednotlivých objektov, napr. meno, farba, rýchlosť a pod.



Príklad

- Vytvára sa hierarchia dedičnosti
- Postupuje sa dvoma smermi:
 - Zdola na hor (zovšeobecnenie) – hľadajú sa triedy so spoločnými vlastnosťami, ktoré sú vybraté do nadtriedy
 - Zdola nahor (špecializácia) – existujúce triedy „zjemňujeme“ pomocou podtried
- Vytvorená hierarchia by nemala byť veľmi hlboká



- Ak chceme využiť polymorfizmus, vytvárame asociácie k rodičovským triedam

Modely tried projektu

- Doménový model tried
 - Výsledok biznis modelovania
- Konceptuálny model tried
 - Výsledok analýzy
- Implementačný model tried
 - Výsledok návrhu (UML) a implementácie (kód)

Ďakujem za pozornosť

Vaše otázky...

