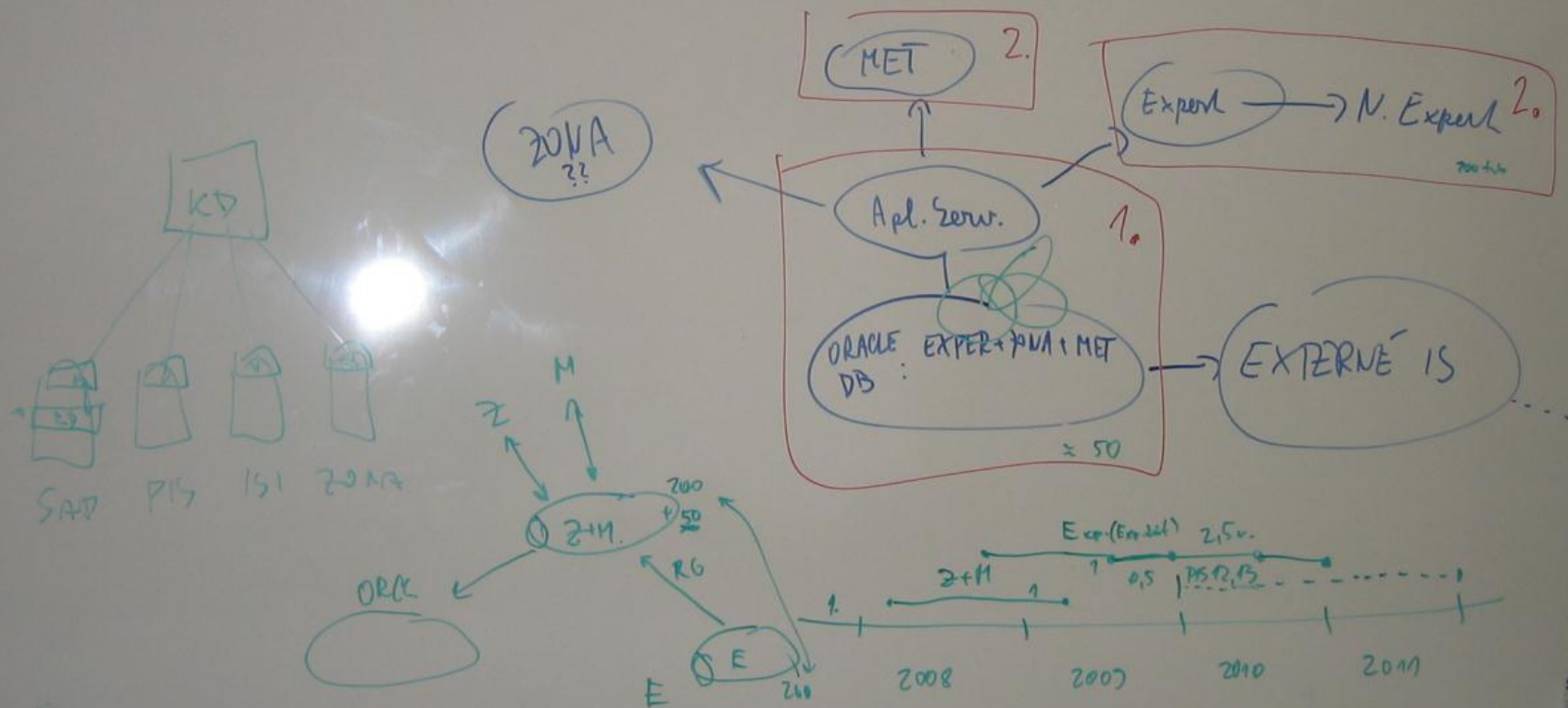


8

Návrh II

- Analýza:
 - Logický model tvoreného systému
 - Analýza požiadaviek z pohľadu problémovej domény
- Návrh
 - Presná špecifikácia spôsobov ako to implementovať
 - Zlúčenie technických riešení
 - Perzistencia objektov
 - Ich distribúcia
 - Architektúra
 - GUI
 - Založený na analytickom modeli

Návrh architektúry



Návrhové modely

- Návrhových podsystémov
- Návrhových tried
- Rozhraní
- Návrhových realizácií prípadov použitia
- Diagramov nasadenia

Návrh tried

Vstupy

- Realizácia prípadov použitia
- Návrhová trieda (načrtnutá)
- Rozhrania (načrtnuté)
- Analytická trieda

Výstupy

- Návrhová trieda (úplná)
- Rozhrania (úplne)

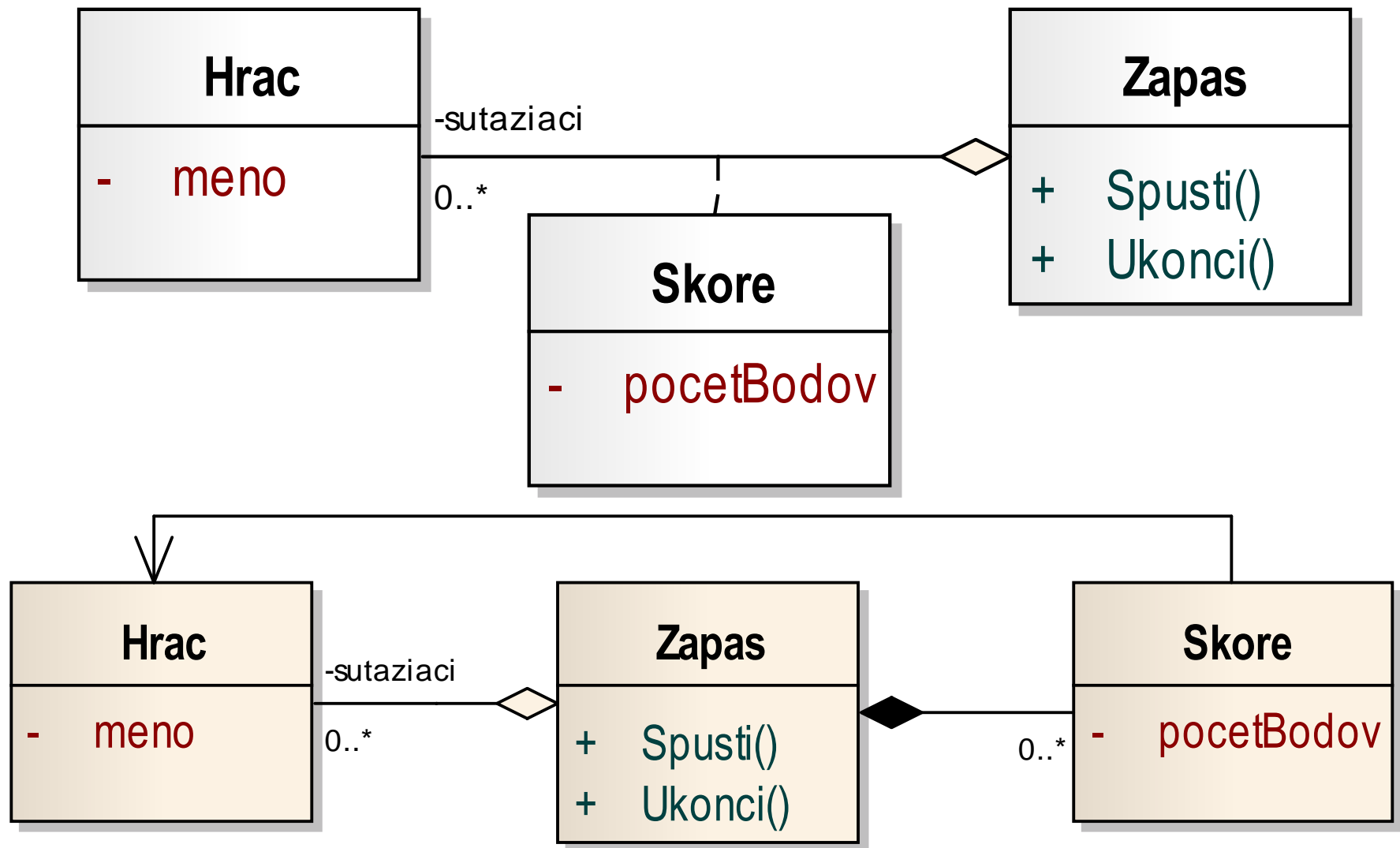
Návrhová trieda – analytická trieda

- Úplná a dostačujúca
- Jednoduchá
- Vysoko súdržná
- Bez tesných väzieb

Objektovo orientovaný návrh

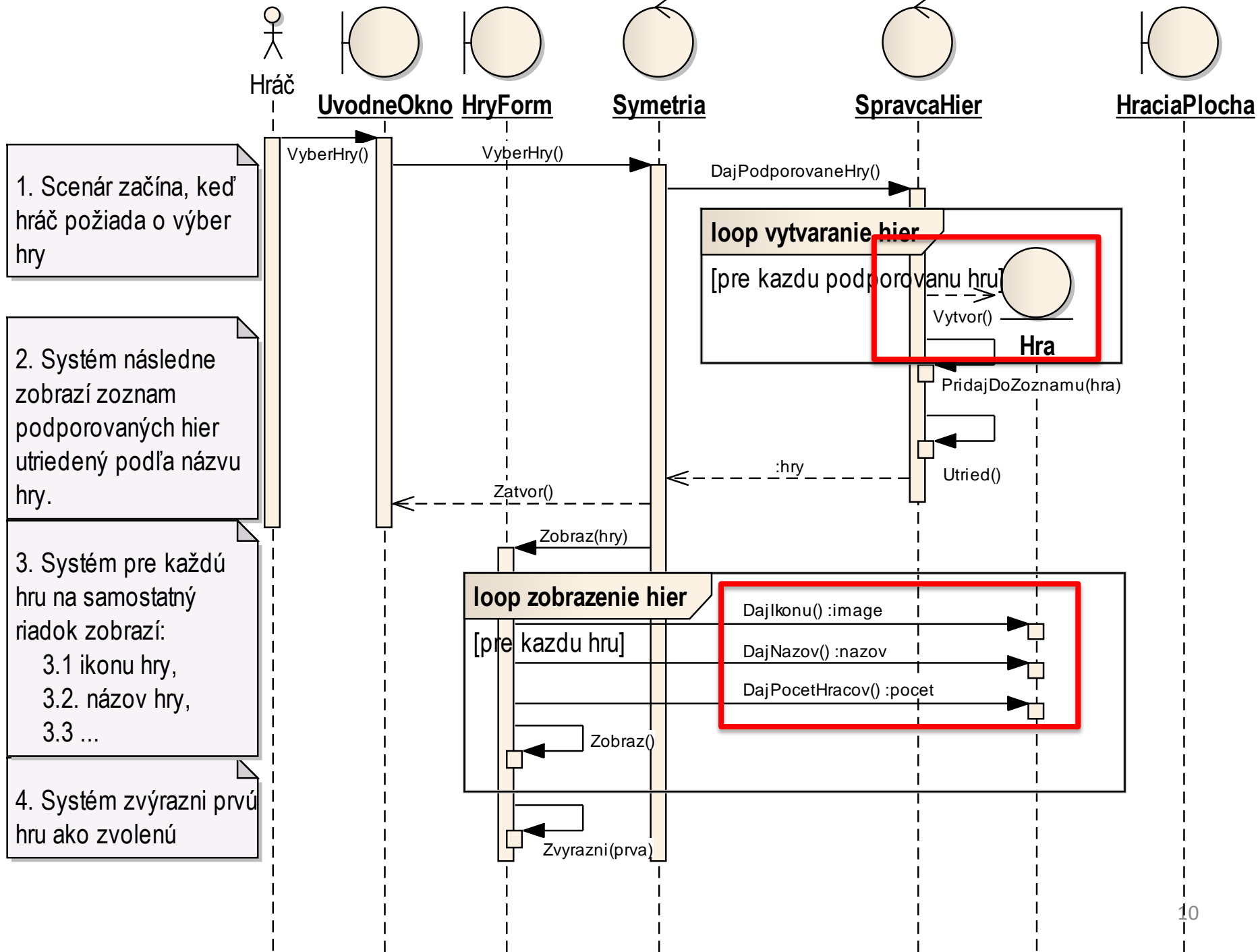
- Vstup: analytické triedy
- Analytická trieda sa môže stať:
 - jedinou triedou,
 - časťou triedy,
 - agregovanou triedou
 - skupinou spriaznených tried,
 - asociáciou a pod.
- Vytvorenie návrhových tried
- Definícia operácií, atribútov
- Definícia asociácií, agregácií a kompozícií

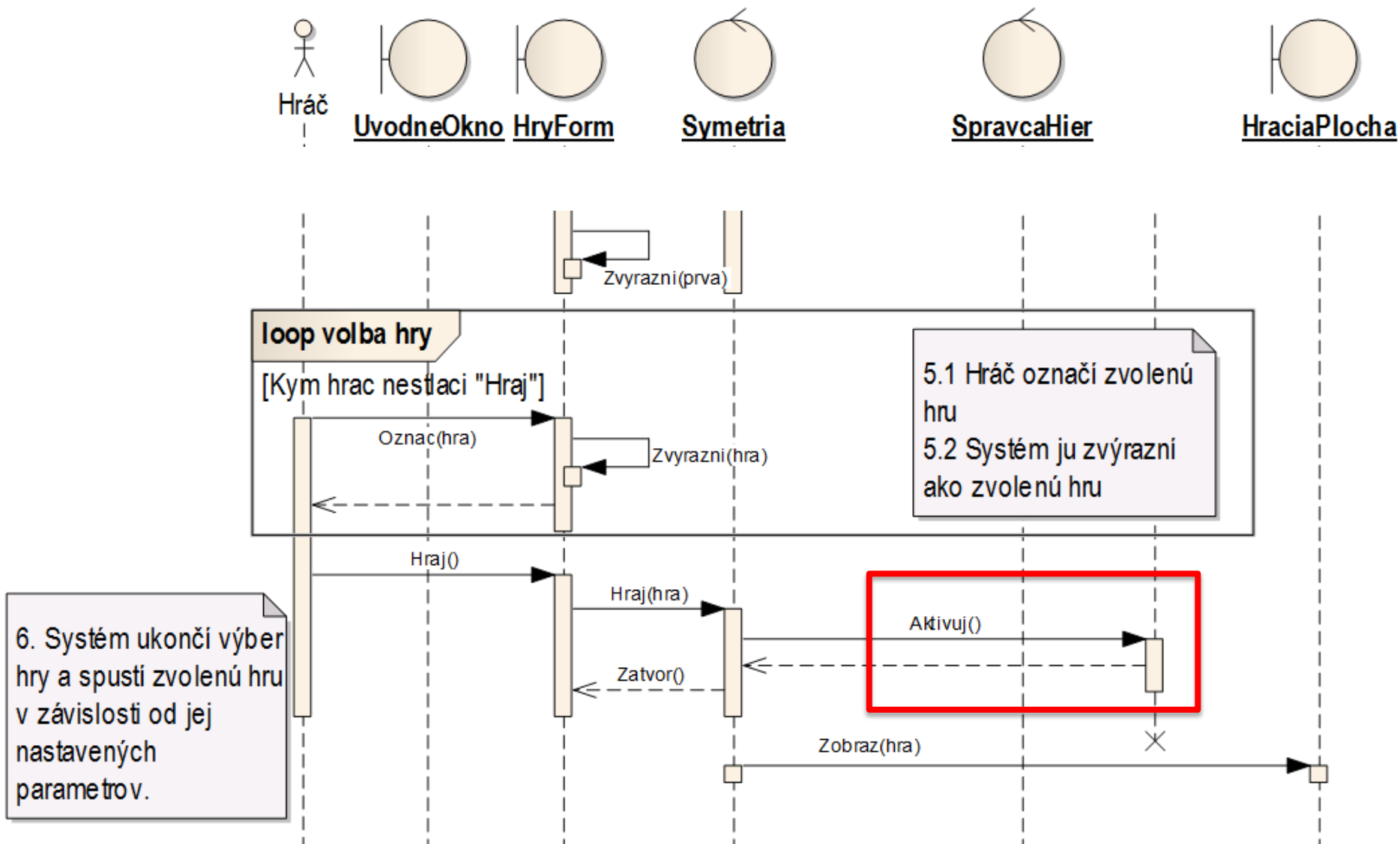
Symetria – zmeny tried



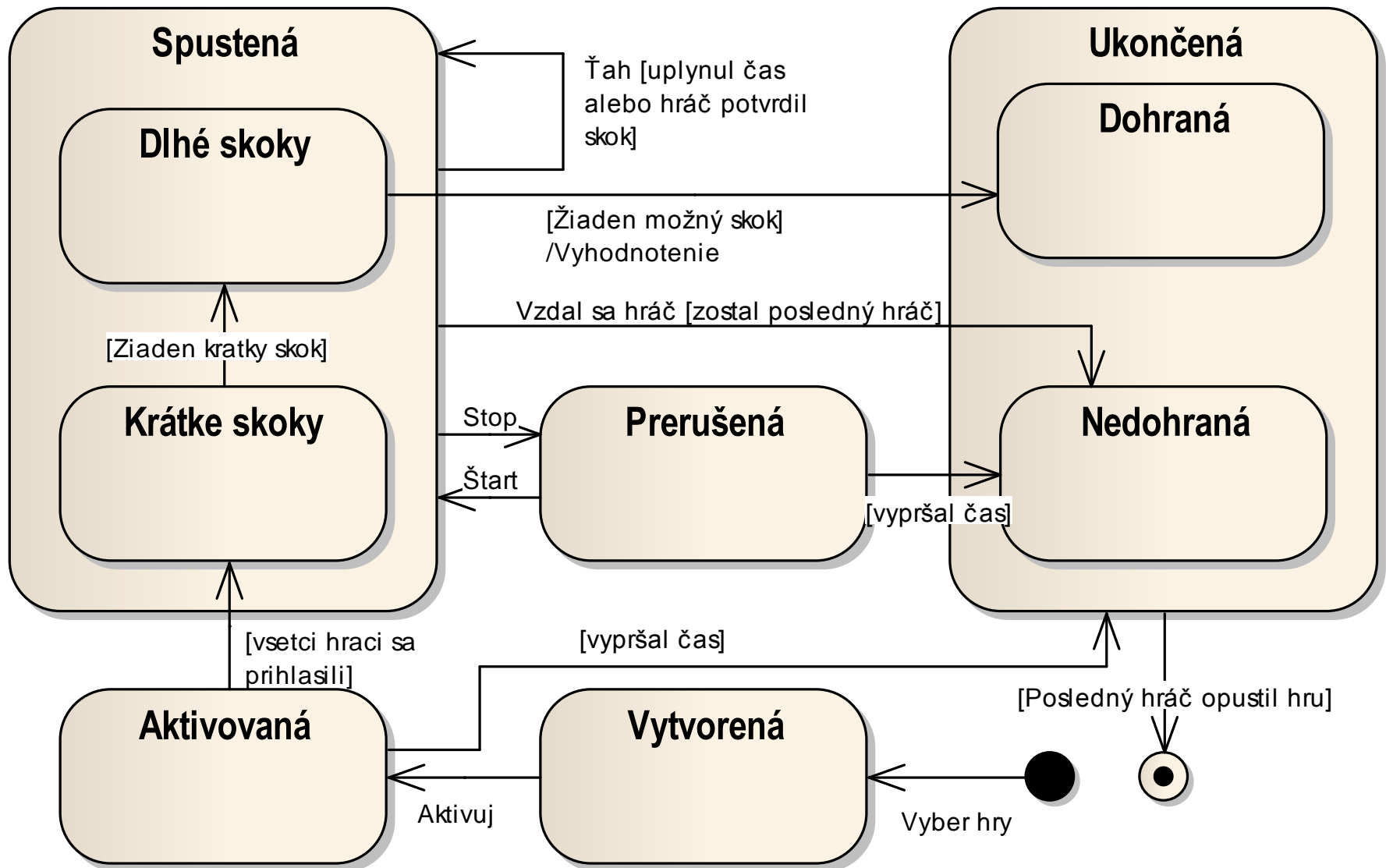
Definícia operácií

- Operácie -zoznam slovies (jednoduchý spôsob)
- Z popisu interakcií medzi objektmi
 - Nakreslenie diagramov spolupráce alebo sekvenčných diagramov
 - Zistenie stimulov, ktoré dokáže objekt prijať –operácie
- Ďalšie možnosti operácií
 - Inicializácia novo vytvorenej inštancie spolu s prepojením s asociovanými objektmi
 - Vytvorenie kópie inštancie
 - Test ekvivalencie inštancií, a pod.
- Operácie popíšeme: názov, parametre, návratová hodnota, krátky popis, viditeľnosť

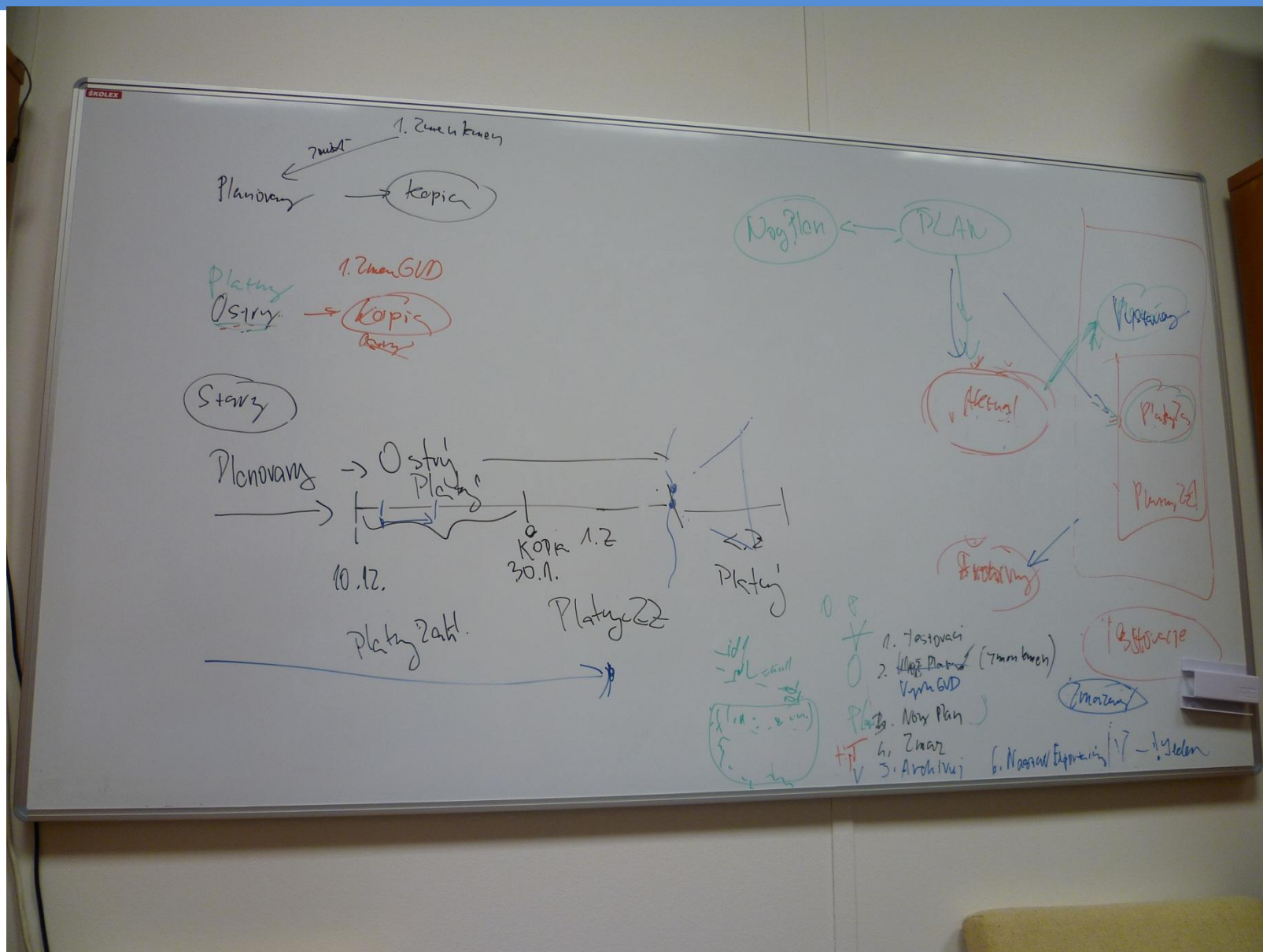




Stavový diagram triedy SkokovaHra



Stavy grafikonu



Doplnenie operácií

SkokovaHra

- + Aktivuj(pocetHracov :byte) : bool
- + DajKonu() : Image
- + DajLimitHracov(max :byte*, min :byte*) : void
- + DajNazov() : String
- DajPocetHracov() : byte
- + HracOdstupil() : bool
- + JeKoniec() : bool
- NastavDlheSkoky() : void
- NastavPocetHracov(pocet :byte) : void
- OverPravidla(typ :TypPravidiel) : bool
- + Pokracuj() : bool
- + SkokovaHra()
- + Spusti() : bool
- + Stop() : bool
- + Ukonci(dohrana :bool) : bool
- + UrobnyTah(tah :Tah) : int
- VypocitajBody(tah :Tah) : int

Definícia atribútov

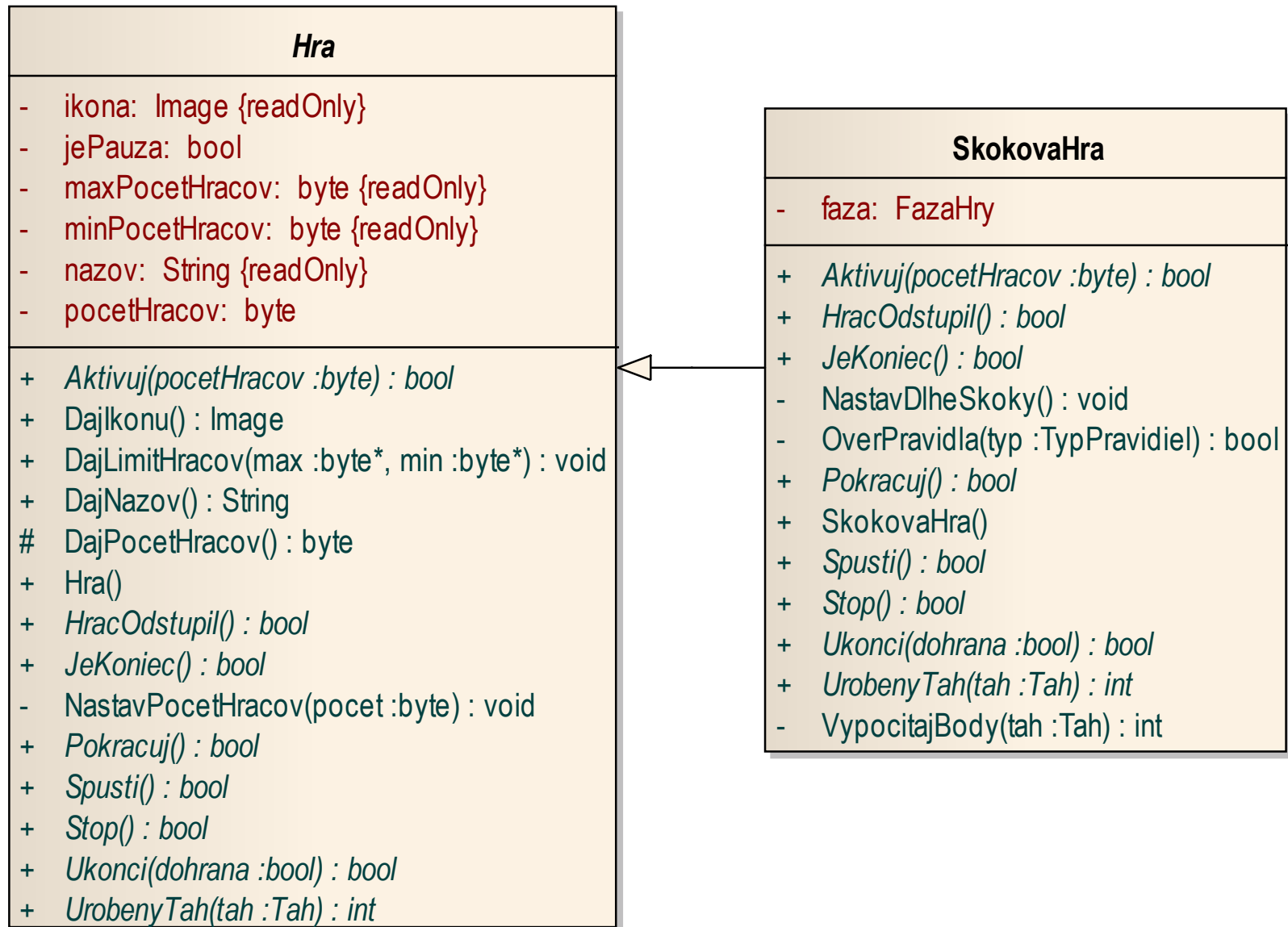
- Varianty:
 - Vychádzame z logických atribútov objektu (čo je potrebné pre zachovanie stavu objektu ?)
 - Aké atribúty sú potrebné pre implementáciu operácií
- Atribúty v návrhu musia byť jednoduché (int, boolean, a pod) alebo musia vyjadrovať hodnotu (string) –inak to budú asociácie
- Atribúty sa popíšu:
 - meno, typ počiatočná hodnota, viditeľnosť
 - Snaha o skrývanie informácií –súkromné atribúty
- Overenie potreby nájdených atribútov

Definícia atribútov

SkokovaHra
<ul style="list-style-type: none">- faza: FazaHry- image: Image {readOnly}- jePauza: bool- maxPocetHracov: byte {readOnly}- minPocetHracov: byte {readOnly}- nazov: String {readOnly}- pocetHracov: byte
<ul style="list-style-type: none">+ Aktivuj(pocetHracov :byte) : bool+ DajIkonu() : Image+ DajLimitHracov(max :byte*, min :byte*) : void+ DajNazov() : String- DajPocetHracov() : byte+ HracOdstupil() : bool+ JeKoniec() : bool- NastavDlheSkoky() : void- NastavPocetHracov(pocet :byte) : void- OverPravidla(typ :TypPravidiel) : bool+ Pokracuj() : bool+ SkokovaHra()+ Spusti() : bool+ StiahniPravidla() : void

- Ak trieda obsahuje viac než 10 atribútov, 10 asociácií alebo 20 operácií
 - zle navrhnutá?
 - je nutné ju rozdeliť?

Doplnenie abstrakcie



Jeden alebo dva modely?

Stratégia	Dôsledky
Spresnenie analytického modelu na návrhový	Jeden návrhový, ale žiaden analytický
Analytický model spresníme na návrhový a použijeme CASE nástroj na obnovu analytického	Jeden návrhový, ale obnovený analytický nemusí byť dostatočný
Ustálime analytický a jeho kópiu spresníme na návrhový	Dva nesynchronizované modely
Udržujeme 2 samostatné modely	Dva synchronizované modely – náročná údržba

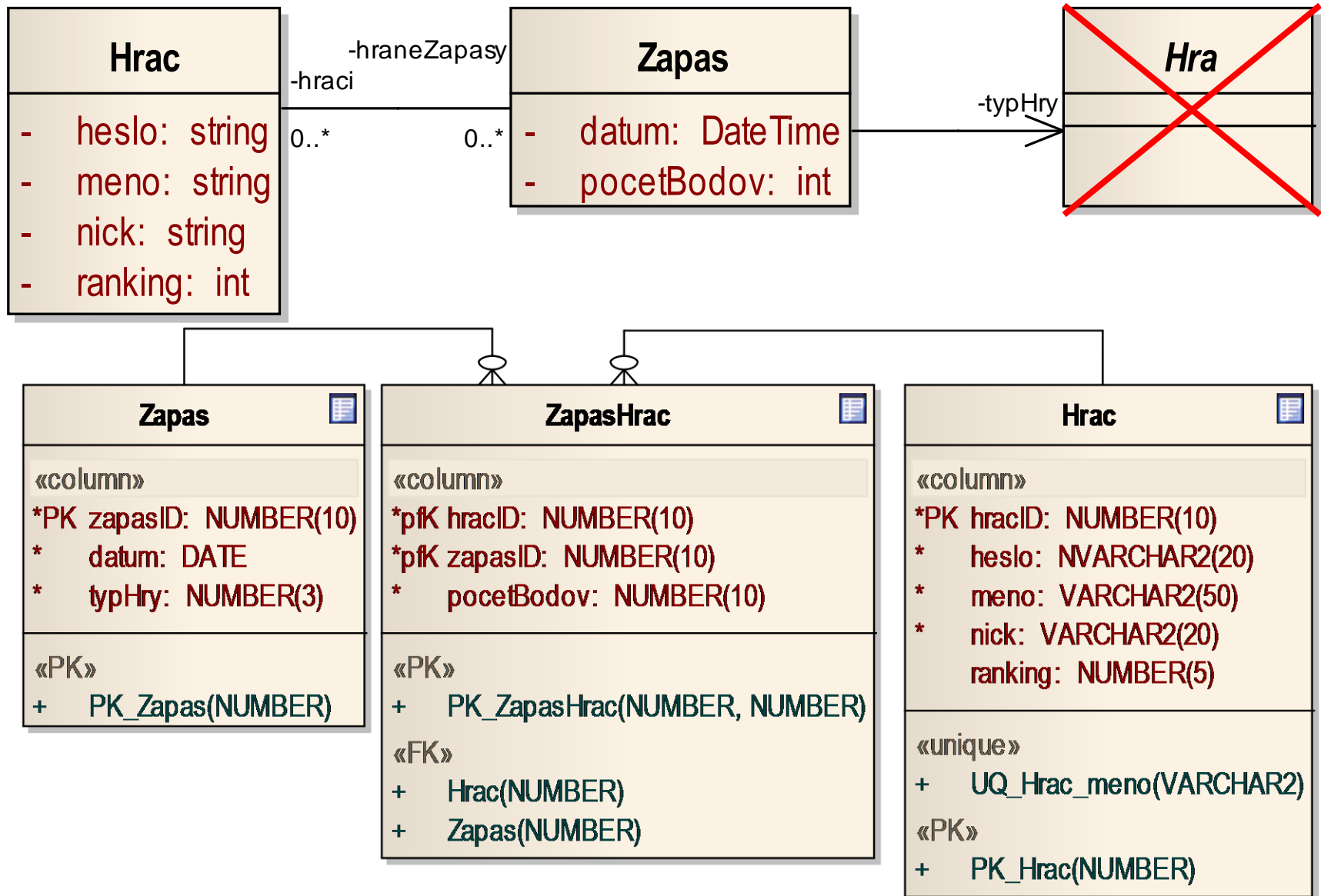
Význam analytického modelu

- Nové osoby v projekte
- Porozumenie systému po dlhej dobe
- Pochopenie systému –uspokojovanie požiadaviek
- Sledovateľnosť požiadaviek
- Plánovanie údržby a rozširovania
- Pochopenie logickej architektúry

Tvorba návrhových tried

- Hraničné triedy
 - Ak sú k dispozícii nástroje pre návrh GUI, potom jedna hraničná trieda = jedno okno alebo formulár
 - Jedna trieda = API alebo protokol
- Entitné (dátové) triedy
 - Často pasívne a perzistentné-implementácia v súbore alebo v relačných databázach
 - Ak nie sú perzistentné–implementácia v pamäti
- Riadiace triedy
 - Obsahujú aplikačnú logiku

Vytvorenie dátového modelu



Realizácia prípadov použitia - návrh

Vstupy

- Model požiadaviek
- Model prípadov použitia
- Model analýzy
- Model návrhu
- Model nasadenia

Výstupy

- Realizácia prípadov použitia - návrh
- Rozhrania (načrtnuté)
- Návrhové triedy (načrtnuté)
- Podsystem (načrtnutý)

- Namiesto analytických tried –návrhové, rozhrania, komponenty
- Odhaľovanie nových nefunkčných požiadaviek a tried
- Identifikácia návrhových vzorov
- Overenie realizácie prípadov použitia
- V návrhu nesmú chýbať správanie potrebné pre niektorý z prípadov použitia

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...

