

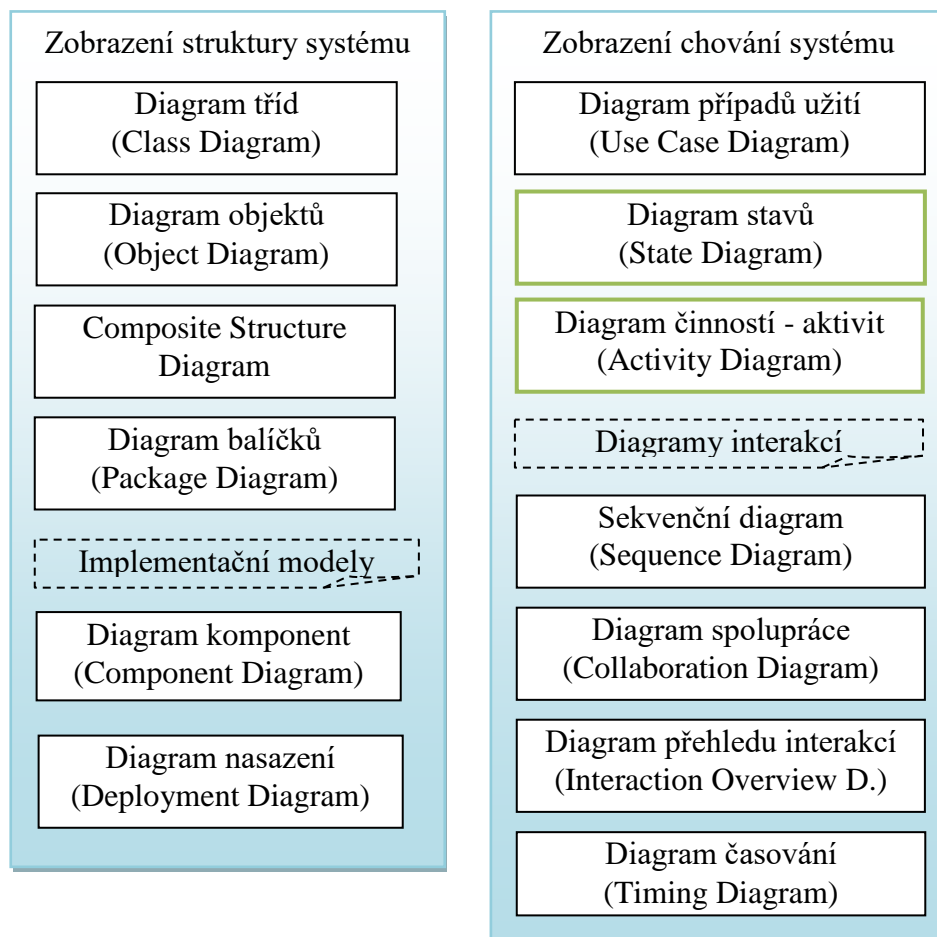
Předmět:	Projektování IS
Téma:	Objektový přístup k vývoji IS (část C)

Vyučující:	dr. Dušan Kajzar	Školní rok:	2020/2021
------------	------------------	-------------	-----------

Obsah:

1. Stavový diagram (State Diagram)..... 2
2. Diagram aktivit / činností (Activity Diagram)..... 9

Kde se nacházíme ?



1. Stavový diagram (State Diagram)

Účel modelu:

- zobrazit množinu **stavů**, kterými jednotlivé **objekty** procházejí v čase,
- zobrazit příčiny **přechodů** mezi stavy,
- pohled na objekt jako na **stavový automat**.

Použití:

- modelování životního cyklu **objektů** (vznik, průběh, zánik),
- modelování dynamiky **subsystémů**,
- popis chování objektu dané třídy (subsystému) v různých stavech.

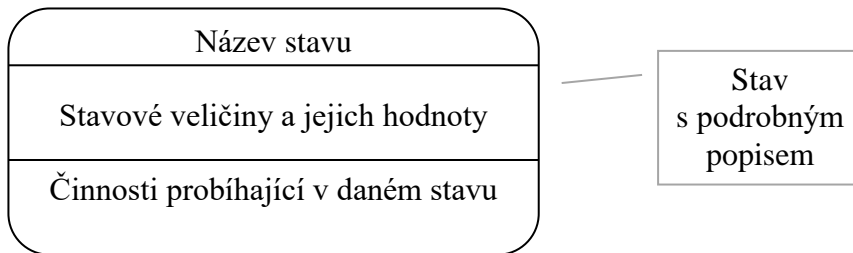
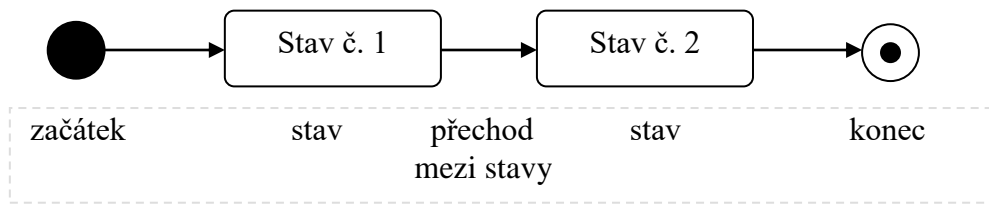
Stavový diagram zobrazuje:

- počáteční a koncový stav každého objektu,
- stavy, do nichž se objekt může dostat,
- podmínky (důvody) přechodů mezi jednotlivými stavy,
- stavové veličiny – hodnoty klíčových atributů v daném stavu,
- činnosti v jednotlivých stavech,
- činnosti při příchodu či opuštění daného stavu,
- akce během přechodu mezi stavy.

Pojem „stav objektu“:

- je situace v životě objektu, která má pro systém nějaký význam,
- během níž objekt splňuje nějakou podmínku,
- stavy objektu rozlišujeme na základě:
 - hodnot vybraných atributů objektu (tzv. stavové veličiny),
 - aktuálně vykonávané aktivity,
 - reakcí (spolupráce) s jinými objekty,
- podrobnost rozlišení stavů:
 - mezi stavy objektu musí být „významově podstatný rozdíl“,
 - tj. aby rozlišení stavů mělo nějaký smysl.

Základní grafické znaky:



Tři základní typy činností probíhajících v daném stavu:

- vstup (enter) – činnosti při vstupu objektu do daného stavu,
- výstup (exit) - činnosti při výstupu objektu z daného stavu,
- provádění (do) - činnosti při pobytu objektu v daném stavu.

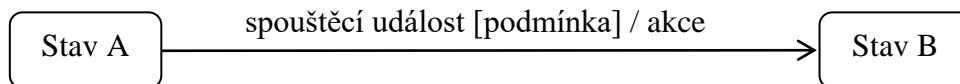
Pseudostavy:

- černý kroužek - začátek diagramu, spojování přechodů,
- terčík – konec diagramu,
- kosočtverec – větvení (pseudostav volby).



Přechody mezi stavy:

- udalost_1, udalost_2 [podmínka] / akce,
- obrázek - spouštěcí událost a akce vykonaná při přechodu mezi stavy.



Důvody přechodu mezi stavy:

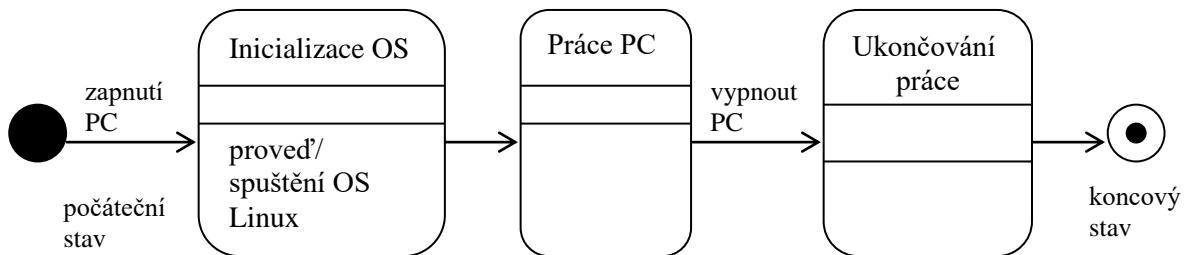
- reakce na určitou událost - výskyt spouštěcí události,
- v důsledku činností probíhajících v daném stavu.

Typy událostí:

- přijetí zprávy (jiný objekt zavolal metodu daného objektu),
- přijetí signálu (informace, dat) – viz dále,
- časová událost,
- objekt indikoval změnu podmínek své existence (vnitřních, vnějších).

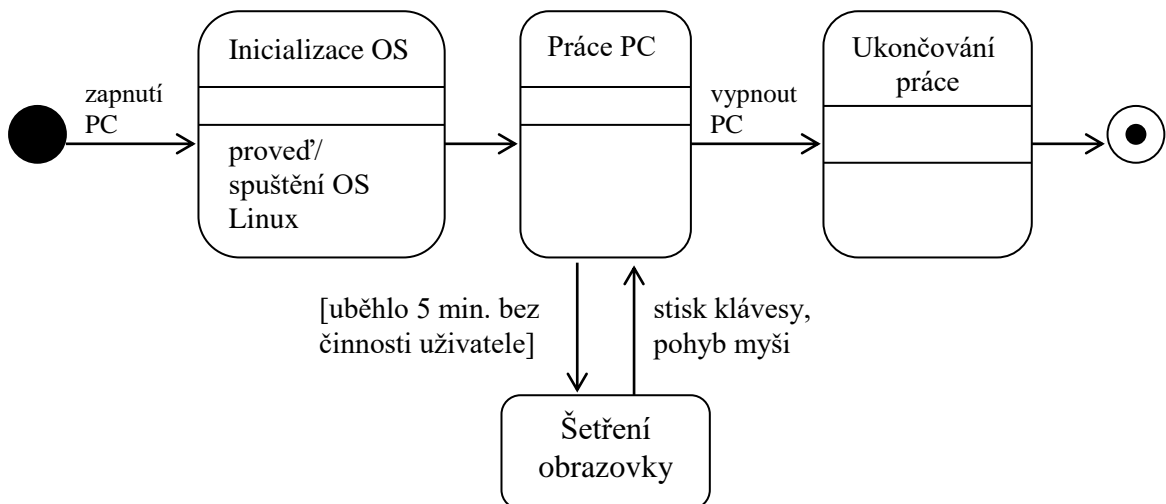
Příklad stavového diagramu:

- obrázek - jednoduchý příklad stavového diagramu.



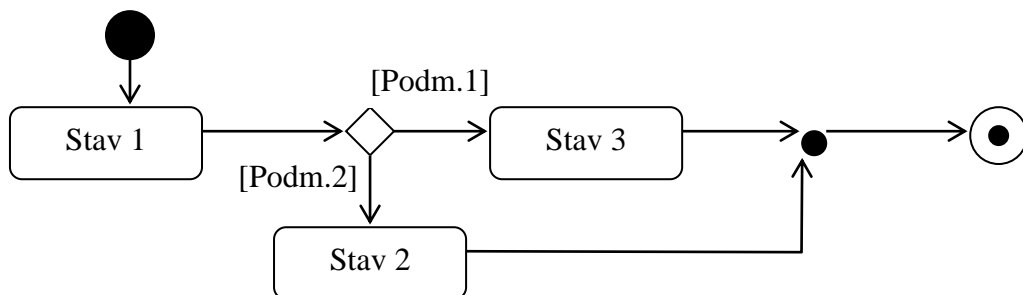
Strážní podmínka:

- obrázek - příklad strážní podmínky.



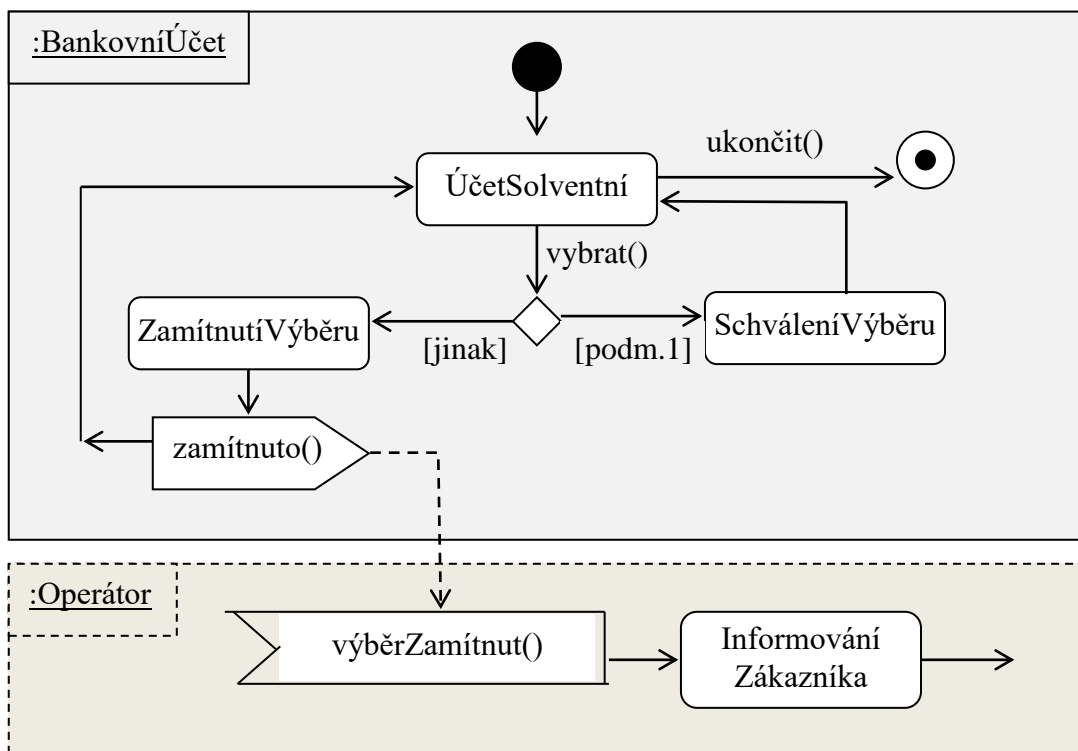
Větvení přechodů:

- obrázek - příklad větvení přechodů mezi stavy.



Signál - odeslání a příjem signálu:

- signál je **jednosměrný asynchronní** komunikační tok mezi objekty,
- nemá operace, pouze přenáší informaci ve svých parametrech,
- ozn. konvexní a konkávní pětiúhelník,
- obrázek - grafické znázornění vyslání a zpracování signálu.



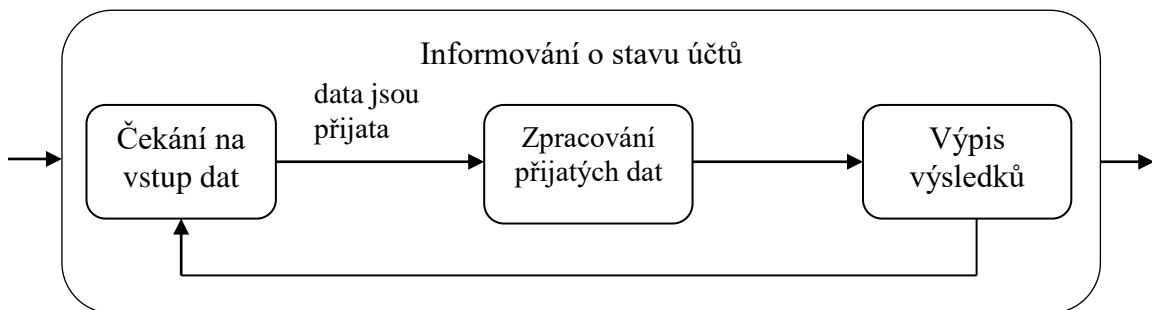
Vnořené podstavy:

- reprezentují vnořený stavový automat,
- vnořený podstav dědí všechny přechody svého mateřského stavu,
- obsahuje-li mateřský stav nějaký přechod, pak všechny podstavy také tento přechod obsahují.

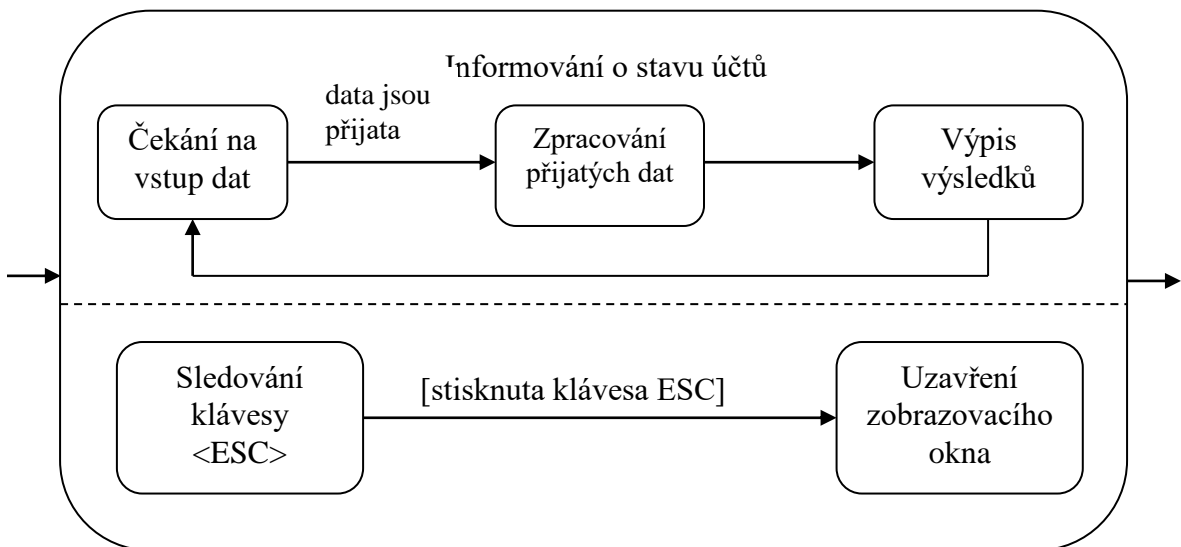
Dva druhy podstavů:

- sekvenční a paralelní (souběžné).

Sekvenční podstavy:



Paralelní podstavy:



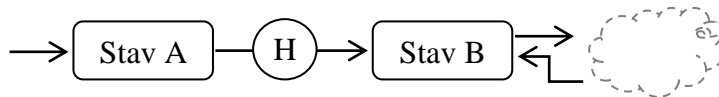
Pseudostavy složeného stavu:

- bílý kroužek
 - místo vstupu do stavu, jenž obsahuje řetězec podstavů,
- terčik - ukončení řetězce podstavů,
- žárovka – ukončení řetězce podstavů
 - místo výstupu z řetězce podstavů externě,
 - tj. mimo nadstav (do zcela jiného stavu objektu),
 - tím se asynchronně ukončí platnost všech řetězců paralelních podstavů.



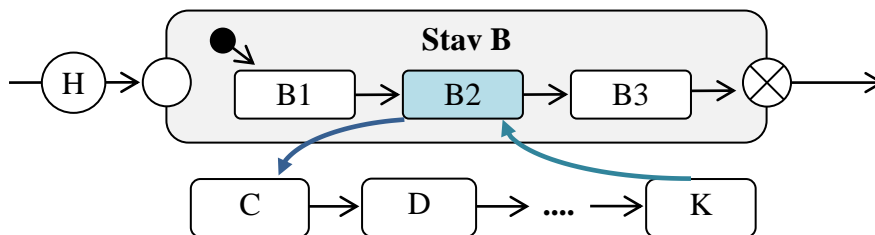
Historie stavu:

- umožňuje po dočasném opuštění složeného stavu opět se vrátit zpět,
- a navíc navázat na poslední podstav, jehož objekt minule dosáhl.



Účel historie stavu:

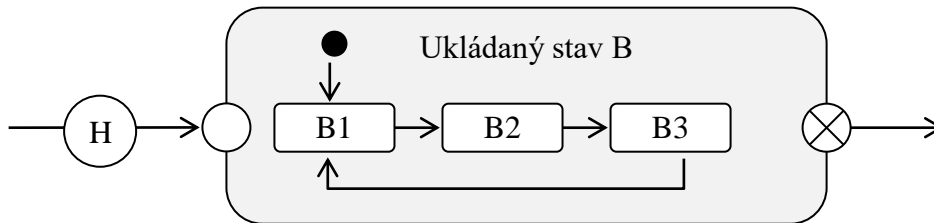
- systém je ve (složeném) stavu B, konkrétně uvnitř vnořeného stavu B2,
 - -> přechází ven ze stavu B (tedy i z B2) do stavu C,
 - -> prochází několika externími stavy C -> D -> ... -> K,
 - -> ze stavu K se vrací do B
- a chceme, aby pokračoval tam, kde minule skončil (tj. v B2).



Indikátor historie (H):

- znamená pamatovat si poslední podstav objektu,
- tj. v jakém podstavu se objekt nacházel v okamžiku opuštění hlavního stavu a hodnoty tohoto podstavu,
- po návratu nastavit objekt na daný pamatovaný podstav,

- jde-li o první vstup do stavu – pak nastavit podstav č.1,
- program musí zajistit ukládání všech potřebných hodnot podstavů, pro možnost návratu,
- hovoříme také o tzv. ukládaném stavu – tj. ukládej aktuální podstavy.



Hluboká historie stavu (H*):

- indikátor znamená pamatovat a přeměřovat vstupy na podstav do libovolné hloubky vnoření podstavů.

Tvorba stavového diagramu:

- 1) stanovíme **stavové veličiny**, které budou jednotlivé stavy charakterizovat,
- 2) stanovíme **hodnoty** (rozmezí hodnot) stavových veličin pro jednotlivé stavy, tj. stanovíme míru podrobností pro rozlišování stavů objektu,
- 3) identifikujeme a popíšeme všechny **možné stavy** daného objektu,
- 4) určíme všechny **možné přechody** mezi stavy daného objektu,
- 5) pro každý přechod mezi stavy určíme **spouštěcí událost** pro tento přechod,
- 6) pro každý přechod mezi stavy určíme, jaké **akce** během daného přechodu proběhnou,
- 7) ...

Shrnutí významu stavového diagramu:

- umožňuje **modelovat změny**, kterými prochází **objekt (subsystém) v čase**,
- pomáhá **pochopit chování** jednotlivých **objektů** obsažených v systému a zobrazených v diagramu tříd (programátoři musí toto chování správně implementovat),
- pomáhá realizačnímu týmu vyvinout IS **reagující** na veškeré podněty (události) tak, jak se očekává.

2. Diagram aktivit / činností (Activity Diagram)

Účel modelu:

- zobrazit **algoritmy** operací (procesů) ve vyvíjeném IS,
- zachytit sekvenci kroků, rozhodování, větvení, opakování,
- jinými slovy – jde o **objektově orientované vývojové diagramy**.

Použití diagramu aktivit:

- **případy užití** (grafické zachycení procesů),
- **algoritmy výpočtů** (operací / metod tříd a rozhraní),
- popisy činností ve **stavovém** diagramu,
- procesy **spolupráce** různých komponent systému,
- znázornění firemních procesů – **workflow** (!).

Vztah mezi diagramem činností a stavovým diagramem:

- stavový diagram
 - zobrazuje stavy jednotlivých objektů, přechody mezi stavy,
- diagram činností
 - popisuje algoritmy činností,
 - např. probíhající ve stavech, při přechodu mezi stavy.

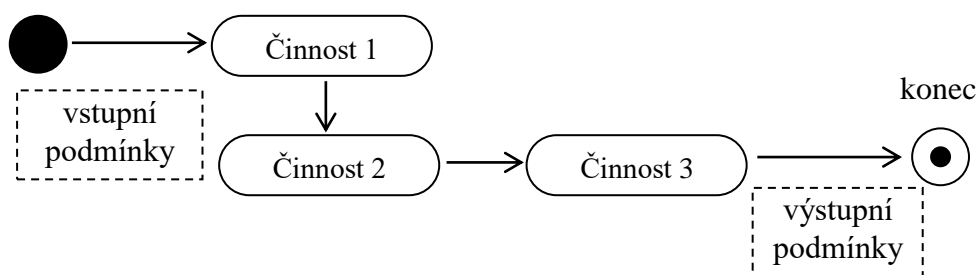
Během modelování:

- vždy mít na paměti, komu je diagram určen,
- tj. k jakému účelu bude sloužit.

Základní grafické znaky diagramu činností:

- obrázek - základní grafické znaky diagramu činností.


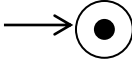
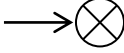
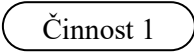
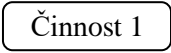
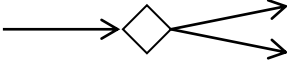
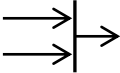
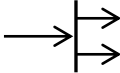
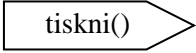

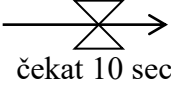
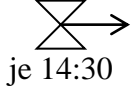
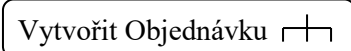
začátek



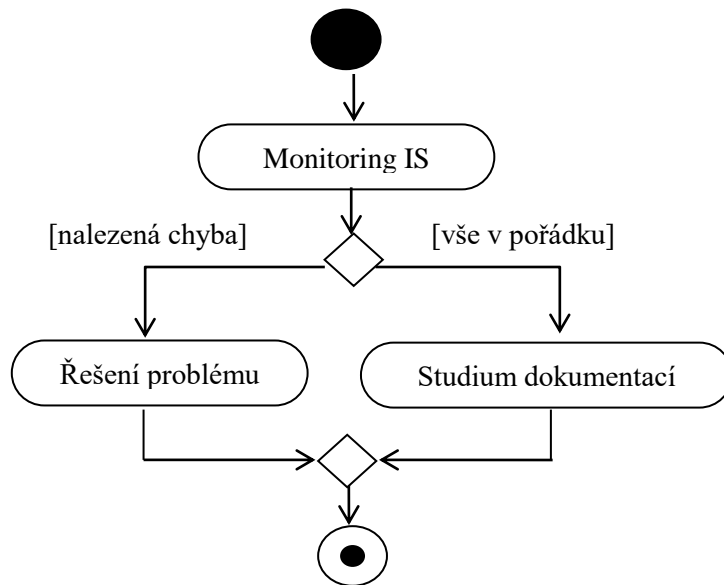
Typy uzlů:

- akční – činnost, aktivita
 - popisují chování, operace, odeslání/přijetí signálu, zpracování události,
- řídicí – řídí postup aktivit
 - začátek, konec, větvení, konec větve, synchronizace činností,
- objektové – zastupují objekty pohybující se v systému
 - předávání hodnot parametrů mezi činnostmi,
 - chovají se jako vyrovnávací paměť pro tokeny,
 - mohou znázorňovat uzly v určitém stavu.

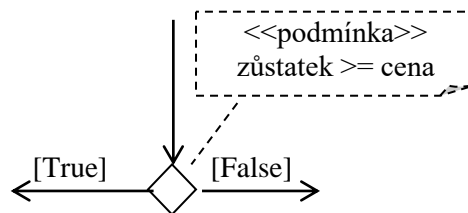
Grafické znaky pro základní typy uzlů:

- začátek a konec aktivity  
- konec cesty (větve) 
- činnosti  
- větvení činností 
- synchronizace činností  
- signály (asynchronní)  
- příjem časové události (reakce na čas)  
- činnost volá další aktivitu (symbol „hrábě“) 

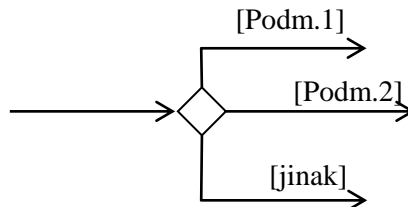
Větvení činností:



Větvení s podmínkou v poznámce:

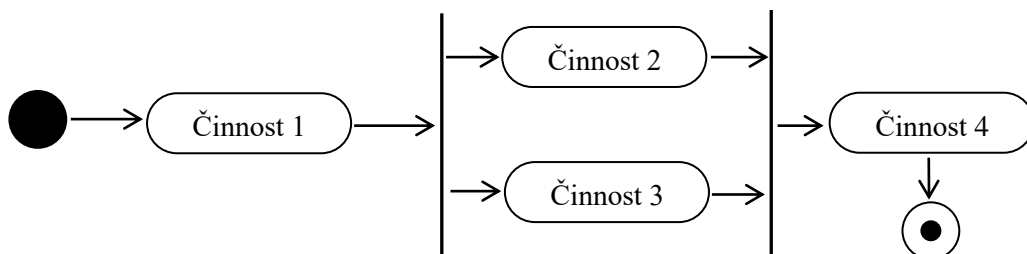


Větvení na více větví:



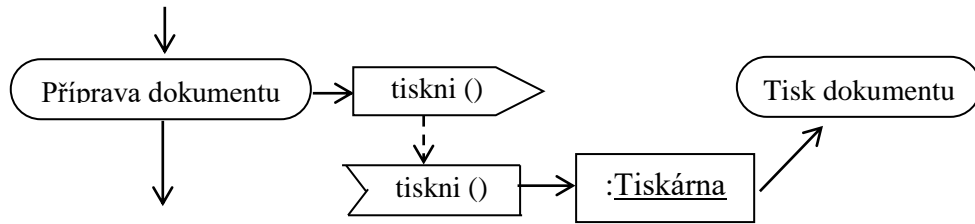
Synchronizace (paralelizmus) činností:

- obrázek - souběžné činnosti a synchronizační čáry.



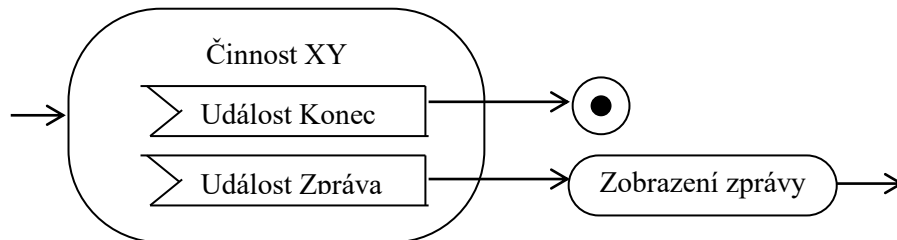
Vyslání a příjem signálu:

- asynchronní komunikace - zaslání a příjem signálu "tiskni()"



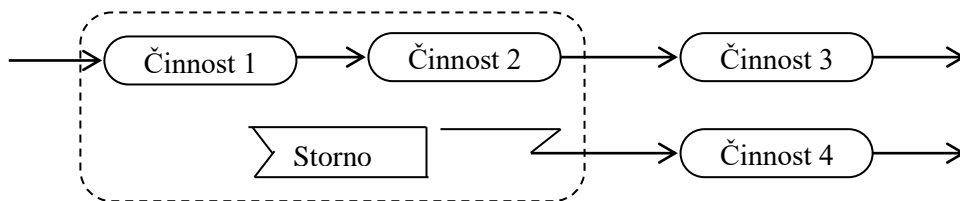
Čekání na událost (signál) + reakce:

- během činnosti očekáváme událost (signál).



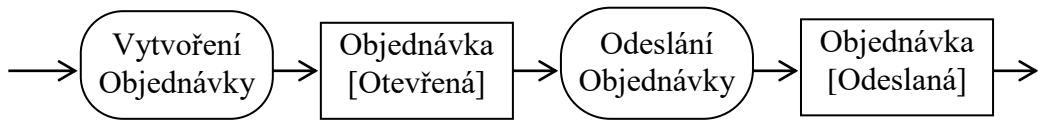
Oblast přerušení:

- signál je očekávaný v průběhu několika činností,
- obrázek - oblast přerušení (inerruptible activity region).



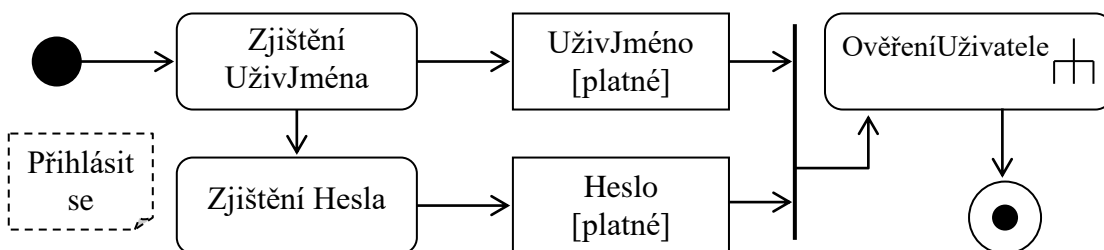
Objektový uzel:

- (viz výše :Tiskárna) signalizuje dostupnost instance dané třídy,
- může znázorňovat vstupní nebo výstupní parametry nějaké aktivity,
- může se chovat jako vyrovnávací paměť
 - token zde čeká, až bude přijat dalším uzlem,
- může znázorňovat uzel v určitém stavu,
- příklad (obrázek) - objednávka ve stavech „otevřená“ a „odeslaná“.

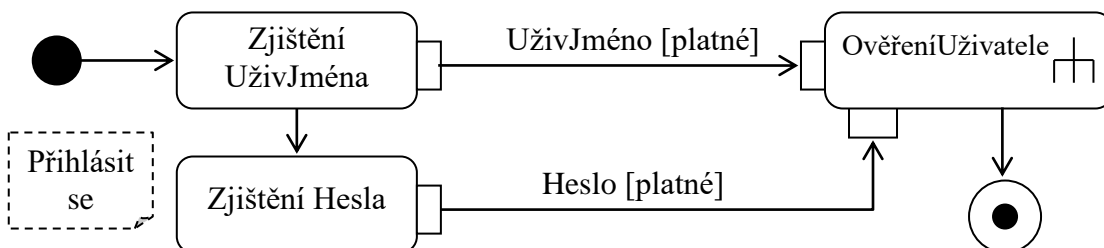


Sponka (pin):

- je objektový uzel zastupující jeden vstup do akce nebo jeden výstup z akce,
- slouží ke zjednodušení diagramu,
- obrázek - aktivita „Přihlásit se“ – s použitím tzv. objektových uzlů,

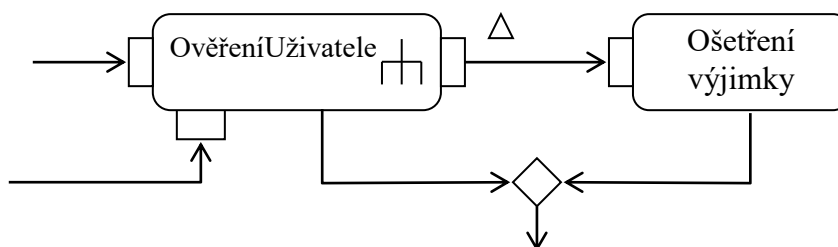


- obrázek - aktivita „Přihlásit se“ - s použitím výstupních a vstupních sponek.



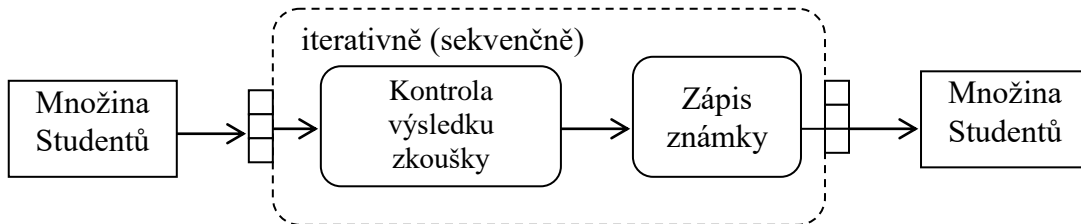
Sponka pro ošetření výjimek:

- pomocí sponky s výjimkou (ozn. trojúhelníčkem),
- obrázek - sponka s ošetřením výjimky.



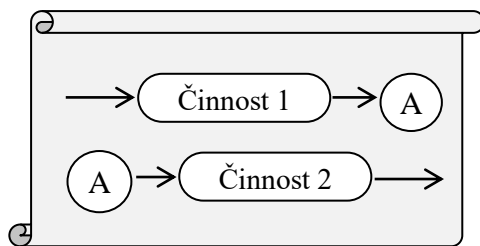
Sponka přijímající množinu objektů:

- přídavný uzel (oblast) – zastupuje množinu objektů,
- tzv. přídavná oblast (čárkovaně) - provede se pro každý prvek množiny,
- režim zpracování množiny prvků - sekvenční, paralelní,
- vstupní a výstupní kolekce musí být téhož typu,
- obrázek - příklad přídavné oblasti.



Spojky:

- značky pro navazování procesů – končí-li papír apod.,
- obrázek - výstupní a vstupní spojka.



Znázornění **rolí objektů** v procesu:

- tzv. zóny odpovědnosti objektů (plavecké dráhy, swimlines),
- kromě objektů (tříd) může jít o
 - organizační jednotky, pracovní role, případy užití,
 - subsystemy, SW komponenty,
 - fyzické počítače (rozdělení procesů na stroje), apod.
- obrázek - role objektů v procesu správy IT.

