

# **Objektově orientované technologie**

## **Cvičení 9 - Tvorba stavového diagramu**

Daniela Szturcová

## 9 Tvorba stavového diagramu

### Cíl cvičení

Nalézt stavy důležitých objektů systému, provázat je přechody a vytvořit stavový diagram.

### 9.1 Teoretický základ

Stavový diagram patří mezi diagramy chování, popisuje životní cyklus vybraného objektu, komponenty, případně celého systému chápaného jako objekt. UML jej definuje jako deterministický automat, což znamená, že pro každý stav musíme nadefinovat podmínky, při kterých automat přejde do stavu jiného. Pokud z jednoho stavu může vést více přechodů, každý z nich je určen jinou podmínkou. Stavový diagram popisuje dynamické vlastnosti objektů systému, reprezentuje všechny stavy, do kterých se popisovaný objekt může dostat během svého životního cyklu. Lze díky němu pochopit chování objektů v různých okamžicích provozu systému.

UML definuje stavový diagram pomocí dále popsaných prvků.

*Stav* vyjadřuje situaci, kdy objekt splňuje nějakou podmínku, provádí nějakou akci nebo čeká na událost. Znázorňujeme jej obdélníkem s oblými rohy. Do obdélníku lze napsat aktivity, které se v daném stavu provádí (mohly být zaznamenány i diagramem aktivit pro daný stav). Například po dosažení věku 18 let, je každý občan označen jako plnoletý, může provádět různé úřední úkony, oženit se/vdát se, volit, získat Řidičák, atd. a toho plynou možnosti jiných stavů občana.

U stavu lze rozlišit aktivitu, která probíhá během trvání stavu. Rozepisujeme pak vstupní, výstupní a interní akce (*entry/vstupniAkce*, *exit/vystupniAkce*, *do/akce*).

*Přechod mezi stavy* představuje spojení mezi dvěma stavy a je vyjádřen orientovanou hranou. Objekt přejde z jednoho stavu do druhého stavu za splnění určitých podmínek. Přechod také může vést z jednoho stavu do téhož stavu.

*Událost*, která způsobí přechod, chápeme jako něco, co se stane v určitém časovém okamžiku, neurčujeme její trvání. Je znázorněna jako doplňující informace u šipky přechodu.

*Podmínky přechodů* Podmínka/y se uvádí v hranatých závorkách přímo u šipky přechodu: [podmínka].

*Akce*, které se během přechodu konají (například výpočet) určují realizaci přechodu.

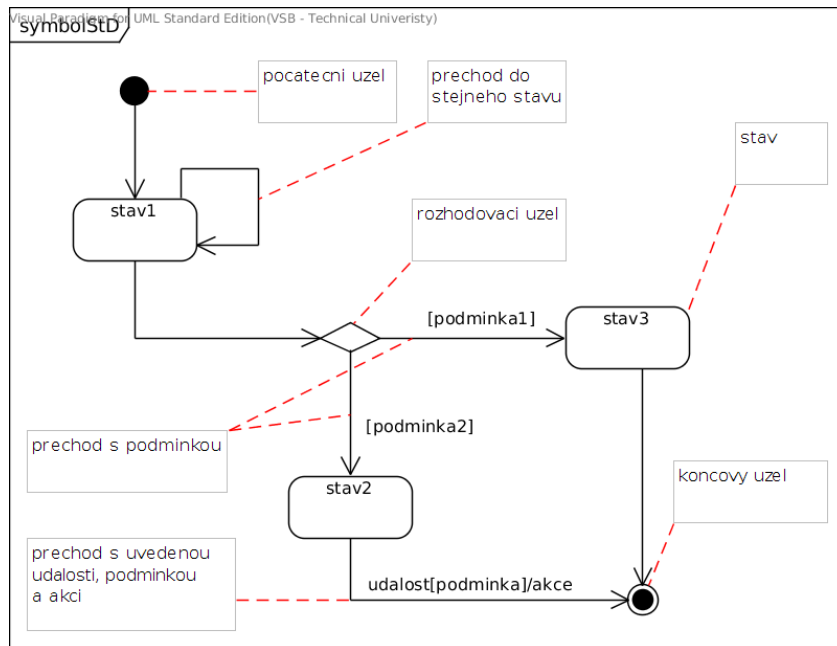
Součástí stavového diagramu je i několik pseudostavů, které jsou chápány jako stavy se speciálním chováním.

*Počáteční bod* je v diagramu pouze jeden. Odtud diagram začíná. Obvykle se z něj přechází bez podmínky k dalšímu stavu, ve kterém se na začátku ocitneme. Je-li podmínka uvedena, musí být automaticky splněna. Z počátku je možný pouze jeden přechod.

*Koncový bod* je v diagramu obvykle jeden, ale podle UML jich může být i více. Je vhodné rozložit tok digramu od levého horního rohu k pravému dolnímu, tj. počáteční bod je nahoře, koncový bod dole.

*Volba* pomáhá rozdělit přechod do více segmentů. Po vyhodnocení volby se přechází do stavu, u jehož segmentu(přechodu) byla splněna strážní podmínka.

Symbolika prvků stavového diagramu je uvedena na obrázku *symbolStd*.



Obrázek 1: Obrázek se symboly stavového diagramu (vytvoreno v akademické licenci Visual Paradigm).

## 9.2 Doporučený postup tvorby diagramu

Při tvorbě stavového diagramu je vhodné postupovat v následujících krocích.

1. Nejprve nalézt významné objekty, ke každému se bude tvořit jeden diagram.
2. Definovat jeho stavy pomocí analýzy hodnot atributů.
3. Zaznamenat stavy do diagramu.
4. Dopsat popis stavu do symbolu stavu.
5. Nalézt přechody mezi stavy, případně přechody s návratem do stejného stavu.
6. Zaznamenat přechody do diagramu.

7. Nalézt omezení u jednotlivých přechodů.
8. Zaznamenat omezení do diagramu.

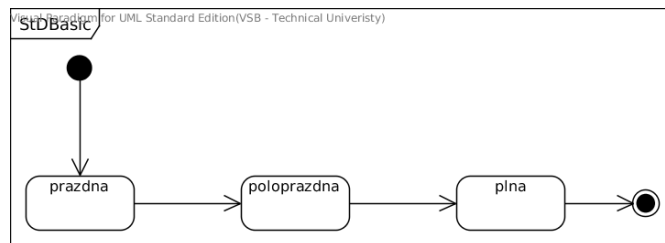
### 9.3 Úkoly

#### Vytváření diagramu

Realizujeme kreslením na tabuli či papíře, nebo použijeme software pro tvorbu diagramů.

1. Za významný objekt modelu nápojového automatu zvolíme Pokladnu.
2. Nadefinujeme stavy Pokladny. Určujícím atributem je počet přijatých mincí. Doména tohoto atributu je od 0 po *maximum* (dáno velikostí pokladny). Je-li počet mincí v pokladně = 0, označíme stav Pokladny *prazdna*.  
Je-li počet mincí v pokladně > 0, označíme stav Pokladny *poloprazdna*.  
Je-li počet mincí v pokladně = *maximum*, označíme stav Pokladny *plna*.
3. Pro každý nalezený stav vytvoříme symbol s odpovídajícím názvem a umístíme do diagramu.
4. Vytvoříme počáteční bod (v některých nástrojích může při založení nového diagramu již být na kreslicí ploše) a z něj zvolíme přechod do stavu *prazdna*.
5. Vyhledáme možné přechody mezi stavy. Ze stavu *prazdna* po přijetí mince dojde ke změně stavu *poloprazdna*. Do stavu *plna* lze přejít pouze vhozením poslední možné mince ze stavu *poloprazdna*.

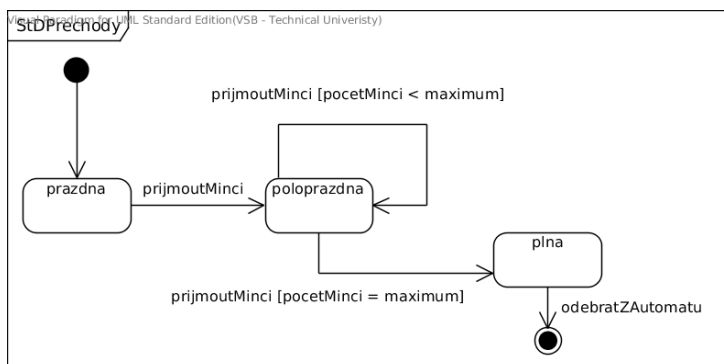
Výsledek záznamu základních přechodů je patrný na obrázku 2.



Obrázek 2: Obrázek základu stavového diagramu (vytvoreno v akademické licenci Visual Paradigm).

6. Dále zde máme přechod s návratem do stejného stavu. Jedná se o případ, kdy Pokladna již obsahuje minci – je *poloprazdna*, další minci přijímá a přijetí nepovede k úplnému zaplnění Pokladny, takže se vrátí zpět do stavu *poloprazdna*.
7. Nadefinujeme omezení u jednotlivých přechodů. U stavu *poloprazdna* mohou nastat dva přechody, protože na základě naplněnosti Pokladny může dojít k přechodu do stejného stavu nebo do stavu *plna*.

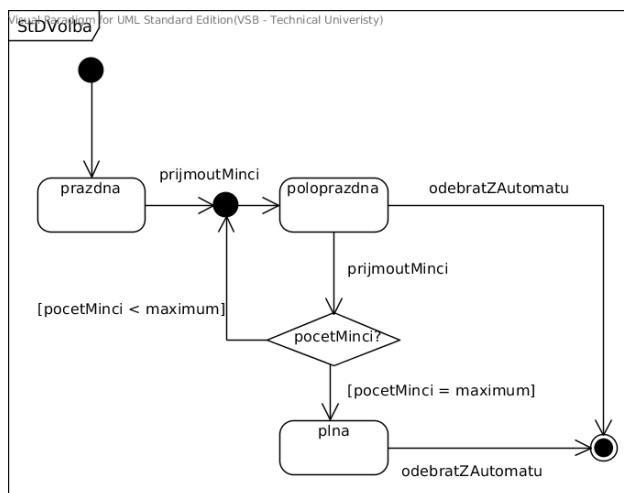
8. Zaznamené události a omezení přechodů do diagramu (viz obrázek 3).



Obrázek 3: Obrázek stavového diagramu s událostmi a podmínkami přechodů (vytvoreno v akademické licenci Visual Paradigm).

9. Vložíme rozhodovací uzel a rozložíme přechod *prijmoutMinci* s podmínkami tak, aby do rozhodovacího uzlu vcházel přechod s událostí. Po vyhodnocení počtu mincí z rozhodovacího uzlu vychází dva segmenty, každý dle strážní podmínky míří do stavu, který je dán ne/splněním podmínky *pocetMinci < maximum*, viz obrázek 4.

10. Upravíme přechody do koncového bodu ze stavů *poloprazdna* a *plna*. Ukončení životnosti Pokladny ve stavu *prazdna* nepředpokládáme.



Obrázek 4: Obrázek po dokončení stavového diagramu pro objekt Pokladna (vytvoreno v akademické licenci Visual Paradigm).

### Samostatná práce

1. Vytvořte stavový diagram pro jiný objekt Nápojového automatu.
2. Vytvořte stavový diagram pro některý objekt systému Taxis.
3. Vytvořte stavový diagram pro objekt Objednavka, která má definovány tyto možné stavy: *nova*, *doplnena*, *Informace*, *dokoncena*, *zrusena*, *zaplacena*. Znáte několik událostí a podmínek přechodů. Po vytvoření objednávky v systému je nutno ji doplnit o další upřesňující informace. Dále je odeslána zákazníkovi žádost o platbu. Ke zrušení objednávky dojde v případě, že zákazník platbu nerealizoval vůbec či do dohodnutého termínu. Po zaplacení objednávky je uzavřena práce s ní a je převedena do stavu *dokoncena*.