

Objektově orientované technologie

Cvičení 6 - Tvorba sekvenčního diagramu

Daniela Szturcová

6 Tvorba sekvenčního diagramu

Cíl cvičení

Pro vybrané případy užití a jejich scénáře detailně zachytit komunikaci mezi objekty.

6.1 Teoretický základ

Sekvenční diagram patří do skupiny diagramů interakcí. Ukazuje, jak objekty komunikují navzájem mezi sebou v časové rovině. Představuje podrobný popis, jak bude realizován určitý scénář, případně návazné scénáře. Jeho smyslem je zachytit postupné zasílání zpráv mezi objekty, zobrazuje tak činnosti několika účastníků během modelované interakce. Představuje spolupráci objektů, která vede ke splnění určité úlohy při užití systému.

Prvky sekvenčního diagramu jsou objekty nebo aktéři (účastníci), jejich čáry života a zprávy, které si mezi sebou posílají.

Čára života reprezentuje účastníka v interakci s ostatními. Označuje se jako buď jako reprezentant třídy - anonymní objekt, případně jako konkrétní instance třídy.

Aktivita účastníka představuje časový interval, kdy je daný účastník aktivní, kdy provádí nějakou činnost, včetně čekání na návrat ze zvané operace. Překrývá čáru života po dobu aktivity účastníka.

Zpráva zobrazuje komunikaci mezi účastníky. Je reprezentována šipkou od jedné čáry života účastníka k další. Její směr představuje od koho (odesílatel) komu (příjemce) je zpráva zaslána. Umístění zpráv shora dolů mezi čarami života postupně vytváří sekvenci zpráv a znázorňuje tak plynutí času shora dolů. Zobrazení návratové zprávy v případech, kdy je zřetelný smysl, není nutno uvádět.

Pro naše potřeby vystačíme s několika základními typy zpráv.

- *Synchronní zpráva* – odesílatel zprávy čeká na odpověď, teprve poté může provádět další akce.
- *Asynchronní zpráva* – odesílatel zprávy nečeká na odpověď, může po jejím odeslání začít s dalšími akcemi.
- *Rekurzivní zpráva* – odesílatel zasílá zprávu sám sobě.

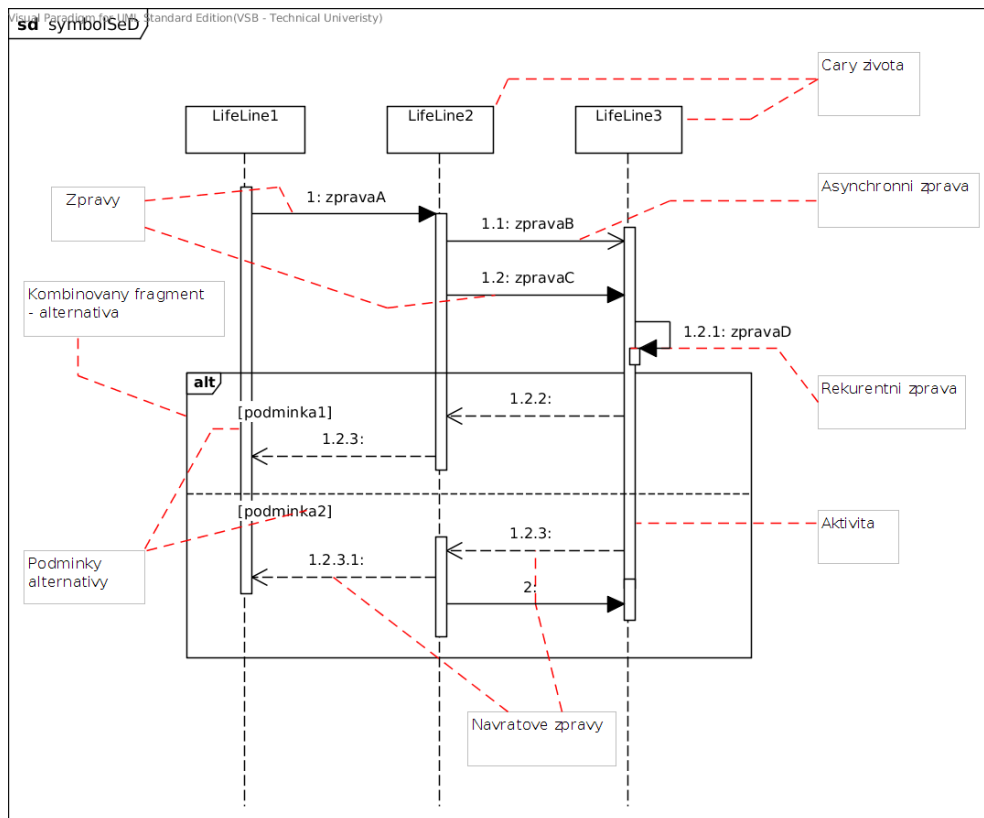
Parametry zpráv, předávaných mezi účastníky, není povinné zapisovat, ale mohou pomoci v při pochopení interakce.

Fragment zahrnuje určitou část interakcí a má speciální význam. Ve většině případů se dělí na více částí, realizace určité části je podmíněna splněním podmínky pro interakce zde definované. Uvádíme několik základních typů.

- *Alt – Alternativa* je fragmentem rozděleným do minimálně dvou částí, kdy interakce proběhnout v té části, kde je splněna podmínka.

- *Opt* – *podmíněný fragment* proběhne jen při splnění podmínky (de facto jednosložková alternativa).
- *Loop* – *smyčka* má definovanou podmínku pro opakovaný průběh obsahu fragmentu.

Zobrazení prvků UML v sekvenčním diagramu je nakresleno na obrázku 1.



Obrázek 1: Obrázek se symboly sekvenčního diagramu (vytvořeno v akademické licenci Visual Paradigm).

6.2 Doporučený postup

Při vytváření sekvenčního diagramu se doporučuje postupovat v následujících krocích.

1. Vybrat si scénář, který bude detailněji rozebrán a definovat zprávy mezi objekty.
2. Rozebrané objekty zaznamenat do sekvenčního diagramu.
3. Podle scénáře doplňovat zprávy podle logické a časové posloupnosti.

4. Kontrolovat konzistenci s třídním diagramem. V případě nesrovnalostí upravit.
5. Upravit zprávy podle typu.

6.3 Úkoly

Vytvoření sekvenčního diagramu

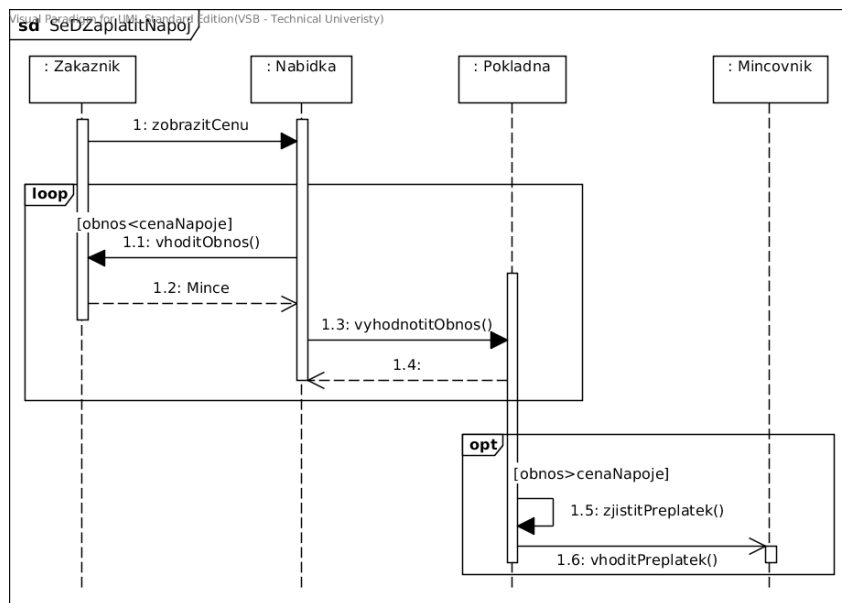
Pro vytváření sekvencí zpráv vezmeme jako základ scénář a z třídního diagramu vybereme ty části struktury, které se účastní realizace scénáře.

1. Vybereme si scénář, který budeme kreslit ve formě sekvenčního diagramu – *ZaplatitNapoj*.
2. V grafickém editoru zvolíme nový sekvenční diagram, nazveme jej SeDZaplatitNapoj a zaznamenáme do něj objekty účastníci se interakcí. (Zakaznik, Nabidka, Pokladna, Mincovnik).
3. Podle scénáře Zakaznik vyžaduje informaci o ceně vybraného nápoje. Odešle tedy zprávu adresovanou Nabidce *zobrazitCenu*. Návratovou hodnotou bude cena vybraného nápoje.
4. Postupně budeme doplňovat zprávy podle logické a časové posloupnosti, vycházíme z nabídky, kterou nám prostředí editoru nabízí u jednotlivých čar života, případně doplníme vlastní názvy zpráv. Pak je ale nutno zajistit konzistenci u tříd v diagramu tříd.
5. Doplníme opakování pro vhadování obnosu, podmínkou je nedostatečná hodnota vhozená Zakaznikem. Podmínku vepíšeme do formuláře, který umožňuje upravovat operand fragmentu *loop*.
6. Doplníme fragment *opt* pro případ, kdy Zakaznik přeplatil a bude mu vrácen přeplatek. Podmínku *hodnotaObnosu > cenaNapoj* umístíme jako omezení operandu fragmentu.
7. Pro zprávu *vhoditPreplatek* upravíme její typ na asynchronní.

Výsledke postupného vytváření sekvenčního diagramu je předveden na obrázku 2.

Samostatná práce

1. Vytvořte sekvenční diagram pro případ užití *ZvolitNapoj*.
2. Vytvořte sekvenční diagram pro vytvořené scénáře systému Taxis.



Obrázek 2: Sekvenční diagramu pro případ užití ZaplatitNapoj (vytvořeno v akademické licenci Visual Paradigm).