

## Zodpovědnost a spolupráce tříd - CRC

CRC = class-responsibility-collaboration

Každá třída má určitou zodpovědnost poskytnout jisté funkce uživateli systému (například aktualizovat nějaký záznam) nebo poskytnout funkce jiné třídě (třeba dodat údaje pro nějaký výpočet). Zodpovědnost určuje, co se od třídy očekává. Je to povinnost třídy poskytnout určitý druh služby.

Cílem techniky používající CRC je rozdělit celkovou funkčnost systému do zodpovědností, které jsou následně přiřazeny jednotlivým třídám. Jakmile známe zodpovědnosti jednotlivých tříd, můžeme zjistit, zda třídy jsou schopny je samy splnit nebo budou potřebovat spolupráci jiných tříd, aby zodpovědnosti splnily.

Dobrý objektově orientovaný návrh je založen na malých soudržných třídách, z nichž každá má omezenou, nicméně dobře definovanou funkčnost. Základní pravidlo je, že žádná třída by neměla mít více než 3 až 4 zodpovědnosti.

Efektivní způsob k nalezení zodpovědností je analýza scénáře událostí. Každý účastník scénáře je zastoupen objektem některé ze tříd systému. Tato skupina objektů následně ztvárňuje události, které se odehrávají ve scénáři. Každá zodpovědnost, která se ve scénáři objeví, je přiřazena nejvhodnějšímu objektu. Pokud některou zodpovědnost nemůže žádný existující objekt splnit, je identifikován nový objekt (a tím i třída). Cílem tohoto postupu je minimalizovat počet a složitost zpráv, které jsou předávány mezi objekty, a vytvořit třídy se zřetelným účelem a vzájemnou soudržností.

Výhoda analýzy takového scénáře je, že se nezabývá podrobnou strukturou jednotlivých objektů (a tím i tříd), čímž lze analyzovat různé možnosti rozdělení zodpovědností mezi třídami bez toho, že by přechod z jedné možnosti na druhou znamenal ztrátu značeného rozsahu již udělané práce.

Následující část ukazuje typický scénář v půjčovně kol. V bodech, kde je identifikována zodpovědnost systému, je v závorce tučným písmem uveden objekt třídy, který ji zajišťuje.

- Eva přijde do půjčovny v pátek jednoho týdne a vybere si horské kolo.
- Pracovnice půjčovny Jana se podívá na evidenční číslo kola a vidí, že toto číslo je 48.
- Jana vloží číslo kola do systému.
- V systému Jana uvidí, že jde skutečně o dámské horské kolo, a zobrazí se jí denní půjčovné 30 Kč a záloha 1000 Kč. (**:Kolo**)
- Eva sdělí, že kolo si chce půjčit na týden.
- Jana tento údaj vloží a zobrazí se jí platba  $210+1000=1210$  Kč. (**:Kolo**)
- Eva souhlasí.
- Jana vloží Evino jméno, adresu a telefonní číslo do systému. (**:Zákazník**)
- Eva zaplatí 1210 Kč.
- Jana zaznamená platbu do systému a systém vytiskne stvrzenku. (**:Platba** ve spolupráci s objektem **Zákazník**)
- Eva souhlasí s tím, že kolo vrátí příští pátek odpoledne.

Zákazník	
Zodpovědnost	Spolupráce
Poskytnout údaje o zákazníkovi. Zachovat historii transakcí. Přidat nové objekty platby.	Zapůjčení Platba

Zapůjčení	
Zodpovědnost	Spolupráce
Poskytnout údaje o zapůjčení. Zajistit úkony spojené s vrácením kola; bylo-li vráceno pozdě, vypočítat a uložit pokutu. Dále uložit částku zaplacenou jako náhrada za poškození.	Kolo

Kolo	
Zodpovědnost	Spolupráce
Poskytnout údaje o kole. Vypočítat půjčovné pro daný počet dní.	

Platba	
Zodpovědnost	Spolupráce
Uložení údajů o celé zákaznickově platbě. Vypočítat a uložit výši vrácené zálohy. Vystavit stvrzenku.	Zapůjčení  Zákazník

## Odvození operací ze zodpovědností

<b>Případ užití:</b> Zapůjčení kola <b>Aktér:</b> Pracovník půjčovny <b>Cíl:</b> Půjčit kolo <b>Přehled událostí:</b> Když zákazník přijde do půjčovny, vybere si kolo. Pracovnice půjčovny nahlédne do systému a sdělí zákazníkovi, jak velké bude půjčovné na daný počet dní. Zákazník zaplatí požadovanou částku, dostane stvrzenku a odejde s kolem. <b>Typický průběh:</b>	
<b>Zákazník, pracovnice půjčovny:</b>	<b>Odezva systému:</b>
1. Zákazník si vybere kolo.	
2. Pracovnice vloží do systému evidenční číslo kola.	3. Zobrazí se údaje o kole včetně denního půjčovného a výše zálohy.
4. Zákazník řekne, na kolik dní chce kolo půjčit.	
5. Pracovnice vloží počet dní do systému.	6. Zobrazí se celková částka za zapůjčení.
7. Zákazník s částkou souhlasí.	
8. Pracovnice vloží údaje o zákazníkovi.	9. Údaje o zákazníkovi se zobrazí.
10. Zákazník zaplatí požadovanou částku.	
11. Pracovnice zaznamená zaplacenou částku.	12. Vytiskne se stvrzenka.
<b>Jiný možný průběh:</b>	
Kroky 8, 9	Zákazník si nepůjčuje kolo poprvé a údaje o něm jsou již uloženy v systému. Pracovnice vloží jen jméno zákazníka a systém údaje o zákazníkovi zobrazí.
Kroky 7 – 12	Zákazník není spokojený s cenou a kolo si nepůjčí.

Jakmile začneme přecházet od analýzy k návrhu, potřebujeme podrobnější údaje, jak zodpovědnost tříd bude uskutečněna. To znamená, že zodpovědnost tříd stanovíme na základě jednotlivých operací a atributů. Musíme zajistit, že každá třída obsahuje údaje a operace potřebné pro zajištění její zodpovědnosti takovým způsobem, jak je uživatel vyžaduje.

Z předchozího přehledu můžeme usoudit, že půjčení kola zahrnuje následující činnosti.

- Potřebujeme mít možnost vložit do systému číslo kola (každé kolo má štítek s evidenčním číslem) a dostat od systému údaje o kole, podle kterých zkontrolujeme, že jde o kolo, které si zákazník vybral, a podle kterých zákazníka informujeme o ceně půjčovného a velikosti zálohy. K tomu potřebujeme operaci *najítKolo*, která má parametr *čísloKola*. Z CRC analýzy víme, že zodpovědnost za poskytnutí informací o kole má třída *Kolo*. Tudíž operace *najítKolo* bude umístěna jako statická operace ve třídě *Kolo*.
- Chceme mít možnost zjistit, kolik bude stát půjčení kola na stanovený počet dní. Zodpovědnost za to má opět třída *Kolo*. Proto v ní bude operace *vypočítatPůjčovné* s parametrem *početDní*.
- V dalším kroku je zapotřebí zaznamenat údaje o zákazníkovi. To je zodpovědností tříd *Zákazník*. Potřebujeme operaci *uložitÚdaje* s parametry *čísloZákazníka*, *jméno*, *adresa*, *telefonníČíslo*. Případně místo operace můžeme k uložení údajů využít konstruktor třídy *Zákazník*, kterým se údaje uloží při vytvoření nového objektu třídy.
- Je vyžadováno, aby systém uložil, co zákazník zaplatil. A to jak půjčovné, tak zálohu. Systém musí vytisknout stvrzenku, případně i pro platbu za více kol. To je zodpovědností třídy *Platba*. V ní potřebujeme operaci *vypočítatCelkovouPlatbu*, která sečte všechny částky půjčovného a uloží je do atributu *celkověZaplaceno*. Dále sečte všechny zálohy a uloží je do atributu *celkověZálohyZaplacené*. Dále potřebujeme operaci *vytisknoutStvrzenku*, která vytiskne stvrzenku v požadovaném tvaru. Jméno zákazníka uvedené na stvrzence tato operace převezme z příslušného objektu třídy *Zákazník*.
- V požadavcích na systém bylo uvedeno, že je zapotřebí zaznamenat údaje o zapůjčení, jako je datum začátku zapůjčení a předpokládaná doba zapůjčení. Ve scénáři se toto neobjevilo. Nicméně je nutné je

do systému zahrnout. Je zodpovědností třídy *Zapůjčení* uložit tyto údaje. Patrně k tomu využijeme konstruktor třídy, který bude mít parametry *datumZačátku* a *početDní*.

V této chvíli máme operace (včetně konstruktorů tříd):

Kolo	najítKolo( <i>čísloKola</i> )
	vypočítatPůjčovné( <i>početDní</i> )
	Kolo()
Zákazník	uložitÚdaje( <i>čísloZákazníka, jméno, adresa, telefonníČíslo</i> )
	Zákazník()
Platba	vypočítatCelkovouPlatbu()
	vytisknoutStvrzenku()
	Platba()
Zapůjčení	Zapůjčení( <i>datumZačátku, početDní</i> )

Nyní bychom obdobným způsobem prošli další scénáře (vrácení kola apod.) a z nich bychom odvodili další operace, které zajišťují funkčnost systému. Konečný výsledek návrhu operací (a atributů):

Zákazník
čísloZákazníka jméno adresa telefonníČíslo
uložitÚdaje( <i>čísloZákazníka, jméno, adresa, telefonníČíslo</i> ) zjistitCoZapůjčeno() <u>najítZákazníka(<i>jméno</i>)</u> editovatÚdaje( <i>jméno, adresa, telefonníČíslo</i> ) údajeZákazníka() Zákazník()

Kolo
čísloKola dostupnost typ velikost provedení model denníPůjčovné záloha
půjčovnéZáloha( <i>početDní</i> ) <u>najítKolo(<i>čísloKola</i>)</u> Kolo()

Speciálníkolo
druh pojištění
najítSpeciálníKolo() SpeciálníKolo()

Zapůjčení
datumZačátku početDní datumVrácení pokutaZaPozdníVrácení náhradaZaPoškození
výšePůjčovného() uložitDatumVrácení( <i>datum</i> ) vypočítatPokutuZaPozdníVrácení() vypočítatNáhraduZaPoškození( <i>poškození</i> ) Zapůjčení( <i>datumZačátku,početDní</i> )

Platba
datumZaplacení celkováPlatba celkověZaplacenéZálohy
vypočítatCelkovouPlatbu() vytisknoutStvrzenku() Platba()