

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## SIMULÁTOR KOMPLEXNÍHO SPLÁTKOVÉHO KALEN- DÁŘE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAL VESELÝ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# SIMULÁTOR KOMPLEXNÍHO SPLÁTKOVÉHO KALEN- DÁŘE

SIMULATION OF COMPLEX INSTALLMENT SCHEDULE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

MICHAL VESELÝ

Ing. JAN SAMEK, Ph.D.

BRNO 2015

## **Abstrakt**

Tato práce se zabývá vztahy z oblasti finanční matematiky týkající se poskytování spotřebitelských a hotovostních úvěrů. Práce se zaměřuje především na problematiku různých typů splácení, úročení a generování splátkových kalendářů. Produktem analýzy těchto matematických principů je simulátor splátkového kalendáře, jenž byl vyvinut pro testovací a konfigurační účely ve společnosti HCI (Home Credit International).

## **Abstract**

The aim of this thesis is to examine relations from the field of financial mathematics concerning consumer and cash loans. Further objects of interest are various types of loan repaying, interest calculation and generating the installment schedule. The result of this mathematical principles analysis is a complex simulator of installment schedule which was created for testing and configuration purposes in HCI company (Home Credit International).

## **Klíčová slova**

finanční matematika, splátkový kalendář, anuita, splácení, úrok, úmor, jistina, úročení, EmbedIT, Home Credit International

## **Keywords**

financial mathematics, installment schedule, annuity, repayment, interest, amortization, principal sum, interest calculation, EmbedIT, Home Credit International

## **Citace**

Michal Veselý: Simulátor komplexního splátkového kalendáře, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Simulátor komplexního splátkového kalendáře

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Samka, Ph.D.

.....  
Michal Veselý  
10. května 2015

## Poděkování

Zde bych velice rád poděkoval vedoucímu mé bakalářské práce, Ing. Janu Samkovi Ph.D., za odborné vedení práce, cenné rady, ochotu a čas. Také bych rád poděkoval společnosti HCI za poskytnutí příjemného technického zázemí.

© Michal Veselý, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
1.1 EmbedIT	3
<b>2 Základní pojmy a principy finanční matematiky</b>	<b>5</b>
2.1 Časová hodnota peněz	5
2.1.1 Budoucí hodnota	5
2.1.2 Současná hodnota	5
2.2 Úrok a úroková sazba	5
2.3 Úmor a jistina	6
2.4 Jednoduché úročení	6
2.4.1 Jednoduché úročení polhůtní (dekursivní)	6
2.4.2 Jednoduché úročení předlhůtní (anticipativní)	7
2.5 Složené úročení	7
2.6 Smíšené úročení	8
2.7 Metody splácení	9
2.7.1 Anuita a anuitní splácení	9
2.7.2 Progresivní splácení	11
2.7.3 Degresivní splácení	13
2.8 Splátkový kalendář	14
<b>3 Návrh simulátoru splátkového kalendáře</b>	<b>15</b>
3.1 Použité technologie	15
3.2 Vstupy simulátoru	15
3.2.1 Výše úvěru (Loan amount)	16
3.2.2 Úroková sazba (Interest rate)	16
3.2.3 Datum podepsání smlouvy a první splátky	16
3.2.4 Typ půjčky (Loan type)	17
3.2.5 Počet splátek (installments)	18
3.2.6 Fixní poplatek (Monthly fee)	18
3.2.7 Výše anuity (Annuity proc)	18
3.2.8 Zaokrouhlení na počet jednotek	21
3.2.9 Zaokrouhlení na desetinná místa (Decimal point rounding)	21
3.3 Vliv zaokrouhlování na generovaný splátkový kalendář	21
3.4 Výstup simulátoru	23

<b>4 Implementace</b>	<b>25</b>
4.1 Popis použitých technologií	25
4.2 Dosažení stanovených cílů	26
4.3 Implementační detaily	26
4.3.1 Vstupní informace	27
4.3.2 Generování splátkového kalendáře	27
4.3.3 Generování výstupních informací	27
4.3.4 Podpůrné moduly	27
<b>5 Testování</b>	<b>29</b>
5.1 Průběh testování funkcionality	29
5.2 Problémy a úskalí při testování	29
5.3 Uživatelské testování	30
5.4 Podpora webovými prohlížeči	30
5.5 Výsledky testování simulátoru	31
5.5.1 Ukázka použitých testů a jejich výsledků	31
<b>6 Závěr</b>	<b>35</b>
6.1 Možná rozšíření	36
<b>A Obsah CD</b>	<b>38</b>

# Kapitola 1

## Úvod

Za několik posledních desítek let přibylo na světovém trhu obrovské množství technologií, na které si moderní společnost natolik zvykla, že už si bez nich naše životy nedovedeme ani představit. Téměř v každé dnešní domácnosti lze nalézt různé technologie, jako jsou mikrovlnná trouba, myčka nádobí, automatická pračka, rychlovarná konvice, každý člen rodiny vlastní mobilní telefon, notebook apod.

Jelikož jsou všechny tyto život zjednodušující technologie poměrně nákladné a rychle se stávají zastaralými, je nutné je jednou za čas obměnit za nové. To ovšem pro každou dnešní rodinu znamená mnohem větší finanční výdaje, než tomu bývalo dříve. Vzhledem k této skutečnosti nenarůstá pouze počet moderních technologií, ale také bank a společností poskytujících spotřebitelské úvěry. Jednou ze společností poskytujících spotřebitelské úvěry je Home Credit International (dále jen HCI), pod kterou spadá i EmbedIT (více v podkapitole 1.1), z jejíž iniciativy vznikla tato práce.

Jelikož problematika poskytování úvěrů zahrnuje spoustu náročných částí, jako volbu typu úročení, určení způsobu splácení a výši jednotlivých splátek, zavádění fixních a procentuálních poplatků, poskytování různých benefitů poctivě platícím klientům a naopak sankcí za nesplácení úvěru atd., je potřeba různých softwarových nástrojů pro zefektivnění správy těchto požadavků.

Tato práce se zabývá analýzou této problematiky a snahou navrhnout jednoduše ovladatelný, funkční a softwarově nezávislý simulátor splátkového kalendáře. Splátkový kalendář si lze nejlépe představit jako tabulku, která zohledňuje všechny atributy pojící se s konkrétním úvěrem a znázorňující, jakým způsobem bude zapůjčená částka v čase splácena, včetně zahrnutí všech přidaných hodnot daných společnostmi poskytující tento úvěr.

Aplikace je vyvíjena pod záštitou části společnosti HCI nazývanou EmbedIT (více informací lze nalézt v podkapitole 1.1) pro jeden z dedikovaných týmů v Brně, který dodává systémové produkty Běloruským zákazníkům.

Hlavním účelem této práce je poskytnout společnosti HCI jednoduchý testovací prostředek pro konfigurování a testování různorodých úvěrových balíčků poskytovaných zákazníkům v Bělorusku.

### 1.1 EmbedIT

EmbedIT je označení části společnosti Home Credit International a.s. (HCI), která poskytuje kompletní IT řešení pro velkou část skupiny PPF - primárně tedy pro celou skupinu Home Credit (viz <http://www.homecredit.net/about-us/group-structure.aspx>) a Air Bank.

Cílem EmbedITu je zajištění komplexní IT infrastruktury včetně nákupu a vybavení HW (servery) a SW (včetně operačních systémů, aplikačních serverů atd.). Nejdůležitější součástí EmbedITu je ale vývojové centrum zajišťující kompletní vývoj ekonomických systémů pro úvěrové produkty včetně integrace s bankovními a karetními (MasterCard, VISA, atd.) systémy.

Tyto systémy jsou vyvíjeny a udržovány pro 10 různých zemí po celém světě (Česká republika, Slovensko, Rusko, Bělorusko, Kazachstán, Čína, Vietnam, Indie, Indonésie, Filipíny), přičemž hlavní vývojové centrum je v Brně. Dále sídlí EmbedIT i v Praze a Ostravě.

Pro jednotlivé země jsou vyvíjeny částečně odlišné verze systémů v rámci dedikovaných týmů, přičemž cílem společnosti je vytvořit jednotné, univerzální a konfigurovatelné jádro informačního ekonomického systému tak, aby jednotlivé země mohli i přes svá specifika toto společné jádro používat. Dedikované týmy jednotlivých zemí se zaměřují pouze na implementaci specifického přizpůsobení základního systému pro danou zemi a vývoj mikro-aplikací specifických pro danou destinaci.



## Kapitola 2

# Základní pojmy a principy finanční matematiky

Tato kapitola je stručným průvodcem základní terminologií finanční matematiky, se kterou se v textu dále pracuje. Obsahuje definice a vysvětlení důležitých pojmů a vztahů z oblasti finanční matematiky, která se zabývá principy úročení, splácení a sestavování splátkového kalendáře.

Podrobnější informace týkající se těchto pojmů lze získat z [1], [2], [3] a [4], odkud jsou také převážně převzaty.

### 2.1 Časová hodnota peněz

Časová hodnota peněz je založena na myšlence, že peníze mají v různém okamžiku různou hodnotu, neboli že hodnota peněz se v průběhu času mění. Princip, resp. metoda časové hodnoty peněz je založena na převodu hodnoty peněz k určitému časovému okamžiku, a to prostřednictvím úročení stanovenou úrokovou sazbou.

Finanční matematika tak převádí hodnotu peněz k určitému časovému okamžiku prostřednictvím úrokové sazby, a to buď k jejich budoucí, nebo současné hodnotě.

#### 2.1.1 Budoucí hodnota

V případě výpočtu budoucí hodnoty se jedná o tzv. úročení. Budoucí hodnota se používá, pokud chceme zjistit, na jakou hodnotu naroste zvolená investice. Nejčastěji je značena  $FV$  (z anglického *future value*) a toto označení budeme také dále používat.

#### 2.1.2 Současná hodnota

Pokud chceme naopak zjistit, kolik musíme nyní uložit na účet, abychom získali v budoucnu určitou částku, nebo jakou hodnotu pro nás má v současnosti určitá investice, mluvíme o současné hodnotě peněz a jedná se o tzv. diskontování. Značí se obvykle  $PV$  (z anglického *present value*).

### 2.2 Úrok a úroková sazba

Úrok je obecně cena za zapůjčení peněz z pohledu dlužníka a odměna za zapůjčení peněz z pohledu věřitele. Výše úroku se udává v peněžních jednotkách a je závislá zejména

na úrokové sazbě.

Úroková sazba je úrok vyjádřený v procentech z hodnoty kapitálu a vždy se vztahuje k určité délce časového období. Rozlišujeme následující časová období:

- roční (značeno p.a.),
- pololetní (značeno p.s.),
- čtvrtletní (značeno p.q.),
- měsíční (značeno p.m.),
- denní (značeno p.d.).

## 2.3 Úmor a jistina

Jistina je základní peněžní částka, která byla půjčena věřitelem.

Úmor je část splátky, která představuje postupné splácení zapůjčené částky. Tento proces se nazývá umořování jistiny.

Součet úmorových částí všech splátek úvěru je roven původně vypůjčené částce (jistině) a součet úrokových částí všech splátek úvěru tvoří odměnu pro věřitele, který jistinu vypůjčil.

## 2.4 Jednoduché úročení

U jednoduchého úročení dochází pouze k úročení původního kapitálu, tudíž nedochází k úročení připsaných úroků z předchozích úrokových období. Jde tedy o využití aritmetické posloupnosti. Tento způsob úročení se obvykle používá, když je počet úrokových období menší než 1.

Jednoduché úročení lze dále dělit na polhůtní (dekursivní) a předlhůtní (anticipativní). Rozdíl mezi nimi spočívá v tom, kdy dochází k úročení a z jaké hodnoty jsou úroky počítány.

### 2.4.1 Jednoduché úročení polhůtní (dekursivní)

U dekursivního úročení dochází k úročení na konci úrokového období a úroky jsou počítány z původní hodnoty kapitálu. Polhůtní úročení je všeobecně známější a typicky se uplatňuje například u bankovních účtů.

Budoucí hodnota kapitálu  $FV$  je rovna současné hodnotě kapitálu  $PV$ , navýšené o připsané úroky  $U$ :

$$FV = PV + U \quad (2.1)$$

Přičemž pro úrok  $U$  platí:

$$U = PV * r * t \quad (2.2)$$

Kde  $r$  je úroková sazba za jedno úrokové období a  $t$  je počet úrokových období.

Dosazením  $U$  do rovnice 2.1 a vytknutím současné hodnoty  $PV$  získáme vztah:

$$FV = PV * (1 + r * t) \quad (2.3)$$

Vztah z rovnice 2.3 předpokládá, že pro každé úrokové období bude platit stejná úroková sazba  $r$ .

V případě, že by se úroková sazba  $r$  měnila pro různá úroková období, musíme pro výpočet budoucí hodnoty kapitálu použít vztah:

$$FV = PV * \left( 1 + \sum_{i=1}^t r_i \right) \quad (2.4)$$

Kde  $r_i$  je úroková sazba v  $i$ -tém úrokovém období.

Z rovnice 2.3 pro výpočet budoucí hodnoty jednoduchého polhůtního úročení lze odvodit, pomocí jednoduchých aritmetických úprav, vztahy pro výpočet ostatních proměnných rovnice:

$$PV = \frac{FV}{1 + r * t} \quad (2.5)$$

$$r = \frac{FV - PV}{PV * t} \quad (2.6)$$

$$t = \frac{FV - PV}{PV * r} \quad (2.7)$$

#### 2.4.2 Jednoduché úročení předlhůtní (anticipativní)

Předlhůtní úročení naopak aplikuje úročení na začátku úrokového období a úroky jsou počítány z budoucí (koncové) hodnoty kapitálu na základě tzv. předlhůtní úrokové sazby  $r_a$ , někdy nazývané *diskontní sazba*. Typicky se anticipativní úročení pojí s diskontováním krátkodobých cenných papírů (především směnek) bankami.

Vztah pro předlhůtní úročení má tvar:

$$PV = FV * (1 - r_a * t) \quad (2.8)$$

Kde  $PV$  je současná hodnota kapitálu (tzv. *aktiva*),  $FV$  odpovídá budoucí hodnotě kapitálu,  $r_a$  je diskontní úroková sazba za jedno úrokové období a  $t$  je počet úrokových období do splatnosti aktiva.

### 2.5 Složené úročení

Složené úročení, na rozdíl od jednoduchého, úročí nejen původní vklad, ale i úroky připsané v předešlých úrokových obdobích. Dochází tak k připisování tzv. úroků z úroků, tudíž při úročení se využívá geometrické posloupnosti. U složeného úročení se aplikuje pouze polhůtní způsob úročení, tudíž úroky se počítají na konci úrokového období a jsou počítány ze současné hodnoty kapitálu.

Obvykle se tento typ úročení používá, pokud je počet úrokových období větší než 1, například u hypotečních úvěrů a podobně.

Následující rovnice znázorňuje vztah pro výpočet budoucí hodnoty kapitálu u složeného úročení:

$$FV = PV * (1 + r)^t \quad (2.9)$$

Kde  $FV$  je budoucí hodnota kapitálu,  $PV$  je současná hodnota kapitálu,  $r$  je úroková sazba za jedno úrokové období a  $t$  je počet úrokových období.

Rovnice 2.9 předpokládá, že úroková sazba bude stejná pro všechna úroková období. Pokud by se úroková sazba v různých úrokových obdobích měnila, budoucí hodnotu kapitálu je nutné počítat následujícím vztahem:

$$FV = PV * \prod_{i=1}^t (1 + r_i) \quad (2.10)$$

Kde  $r_i$  je úroková sazba v  $i$ -tém úrokovém období.

Ze základního vztahu rovnice 2.9 pro výpočet budoucí hodnoty kapitálu lze vyjádřit vztahy pro výpočet úrokové sazby, počtu úrokových období a současné hodnoty kapitálu:

$$r = \sqrt[t]{\frac{FV}{PV}} - 1 \quad (2.11)$$

$$t = \frac{\ln FV - \ln PV}{\ln(1 + r)} \quad (2.12)$$

$$PV = \frac{FV}{(1 + r)^t} \quad (2.13)$$

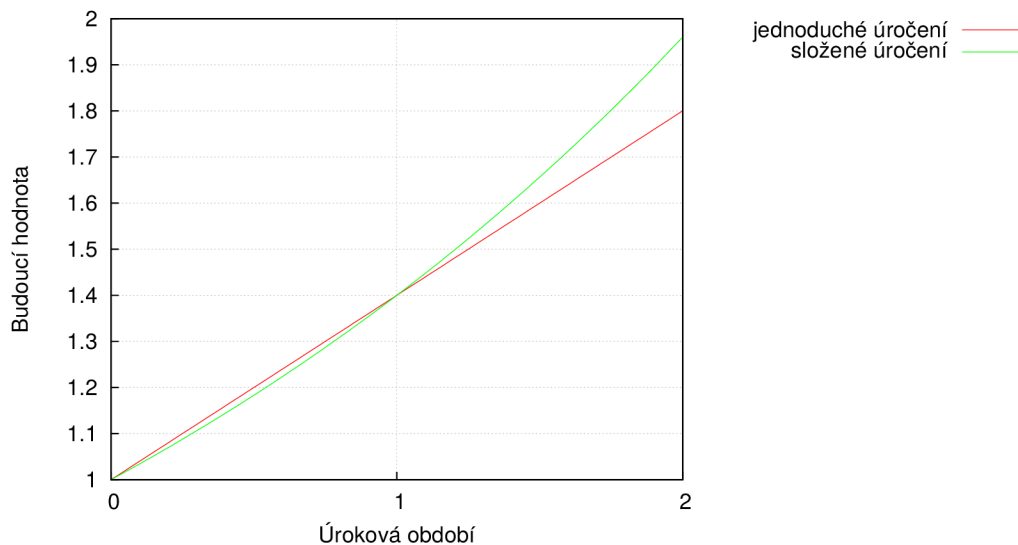
## 2.6 Smíšené úročení

Smíšené úročení je kombinací jednoduchého a složeného úročení. Používá se v případě, kdy v rámci složeného úročení je kapitál úročen necelý počet úrokových období.

Vztah pro smíšené úročení pak vychází z principu, že u celé části počtu úrokových období (značeno  $n$ ) je aplikováno složené úročení a u neceločíselného zbytku počtu úrokových období (značeno  $l$ ) je použito jednoduché úročení. Tento vztah je popsán následující rovnicí:

$$FV = PV * (1 + r)^n * (1 + r * l) \quad (2.14)$$

Smíšené úročení bývá aplikováno, jelikož pro necelé úrokové období dochází k většímu růstu úroků jednoduchým úročením nežli složeným, a proto je ze strany věřitele výhodnější. Tento jev je zachycen v grafu 2.1, ze kterého lze vidět, že úrok roste během prvního úrokového období rychleji jednoduchým úročením a poté už je růst úroku větší u složeného úročení. Z grafu 2.1 lze také vyzorovat, že výše úroku po právě jednom úrokovém období je oběma typy úročení stejná.



Obrázek 2.1: Rozdíl růstu budoucí hodnoty v čase při složeném a jednoduchém úročení.

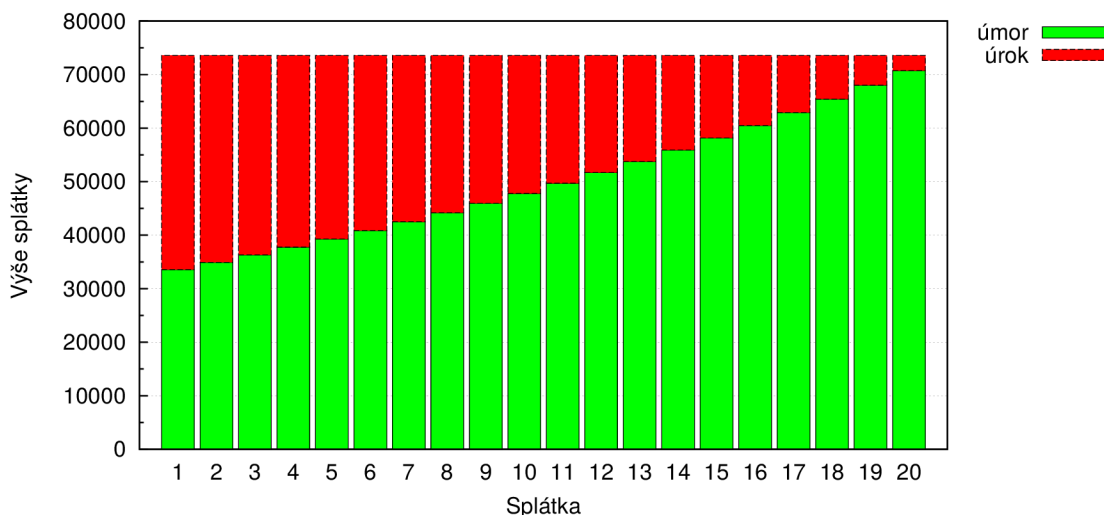
## 2.7 Metody splácení

Nejběžnější a v současnosti nejvíce používanou metodou je anuitní splácení, při níž je výše každé splátky konstantní a v čase neměnná (2.7.1). Existují ovšem i další metody splácení, jejichž princip spočívá v konstantním zvyšování nebo naopak snižování výše splátky pro jednotlivá úroková období. V případě zvyšování jednotlivých splátek mluvíme o progresivním splácení (2.7.2), a v opačném případě, kdy se výše splátek časem snižuje, se jedná o degresivní splácení (2.7.3).

### 2.7.1 Anuita a anuitní splácení

Anuita neboli konstantní splátka představuje stálou platbu hrazenou v pravidelných časových intervalech po dané období. Takový způsob splácení je nazýván anuitní splácení a je typický pro hrazení hypotečních a spotřebitelských úvěrů. Anuita se vždy skládá ze součtu dvou částí, úmoru (viz podkapitola 2.2) a úroku (viz podkapitola 2.3).

Výše anuity je v čase neměnná. Výše úroku obsaženého v anuitě se časem snižuje a výše úmoru se naopak zvyšuje (jelikož postupným umořováním jistiny dochází ke snižování částky, ze které se počítá úrok). Vývoj poměru mezi úrokovou a úmorovou částí splátky lze vidět na obrázku 2.2.



Obrázek 2.2: Vývoj anuity a poměru úrokové a úmorové části půjčky v čase u anuitního splácení.  $PV = 1\,000\,000$ ,  $t = 20$ ,  $r = 4\%$ .

### Výpočet anuitního splácení

Uvažujme zapůjčenou částku  $PV$ , úrokovou sazbu pro jedno úrokové období  $r$ , výši jedné splátky  $a$ , s tím, že dluh bude splacen právě po  $t$  uhrazených splátkách. V takovém případě platí vztah:

$$\left( \left( \left( PV * (r + 1) - a_1 \right) * (r + 1) - a_2 \right) * \dots \right) * (r + 1) - a_t = 0 \quad (2.15)$$

Kde  $a_i$  odpovídá anuitě v  $i$ -tém úrokovém období ( $a_i$  je stejná pro každé  $i \in 1..t$ ).

Roznásobením výrazu z rovnice 2.15 a vyjádřením neznámé  $a$  získáme vztah pro výpočet anuity u anuitního splácení. Tento vztah nám říká, jak vysoká musí být jedna anuitní splátka, aby byl dluh  $PV$ , který je úročen úrokovou sazbou  $r$ , uhrazen právě po  $t$  splátkách.

$$a = \frac{PV * (1 + r)^t}{\sum_{i=0}^{t-1} (1 + r)^i} \quad (2.16)$$

Aritmetickými úpravami lze z rovnice 2.16 získat rovnici pro výpočet původní zapůjčené částky  $PV$  za předpokladu, že známe výši jedné splátky  $a$ :

$$PV = \frac{a * \sum_{i=0}^{t-1} (1 + r)^i}{(1 + r)^t} = a * \sum_{i=1}^t (1 + r)^{-i} \quad (2.17)$$

Rovnice 2.16 a 2.17 ovšem předpokládají, že pro každé úrokové období máme stejnou úrokovou sazbu, což nemusí být vždy pravdou. Pokud zahrneme do rovnic 2.16 a 2.17 předpoklad, že úrokové sazby v jednotlivých úrokových obdobích mohou být různé a  $r_i$

budeme značit úrokovou sazbu pro  $i$ -té úrokové období, získáme rovnice:

$$a = \frac{PV * \prod_{i=1}^t (1 + r_i)}{1 + \sum_{i=2}^t \prod_{k=i}^t (1 + r_k)} \quad (2.18)$$

$$PV = \frac{a * \left(1 + \sum_{i=2}^t \prod_{k=i}^t (1 + r_k)\right)}{\prod_{i=1}^t (1 + r_i)} \quad (2.19)$$

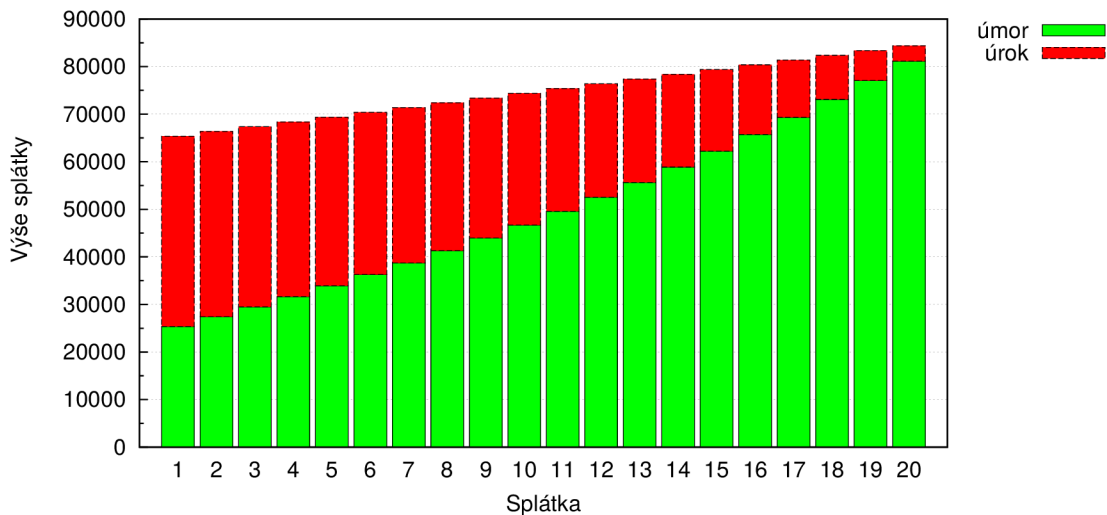
### 2.7.2 Progresivní splácení

Jak již bylo zmíněno, progresivní splácení je takové splácení, při němž v každém úrokovém období dojde k navýšení hodnoty splátky o předem stanovenou konstantu.

Takové splácení je typicky požadováno mladými klienty, kteří si nemohou na počátku splácení úvěru dovolit splácet vysoké částky, ovšem předpokládají, že se jejich finanční situace bude postupem času zlepšovat.

Největší nevýhodou takového splácení je pomalé umořování jistiny v prvních úrokových obdobích, ve kterých nejvíce roste úroková část, a tudíž dlužník uhradí výrazně větší obnos, než by uhradil u stejného úvěru s anuitním nebo degresivním splácením. Na druhou stranu si takový úvěr mohou dovolit i klienti, kteří by si stejný úvěr s jinou metodou splácení dovolit nemohli. Zároveň může dojít k nepříjemnostem, pokud k předpokládanému růstu finančního zabezpečení u klienta nedojde dle jeho původních představ.

Průběh progresivního splácení lze vidět na obrázku 2.3.



Obrázek 2.3: Vývoj výše splátky a poměru úrokové a úmorové části půjčky v čase u progresivního splácení.  $PV = 1\,000\,000$ ,  $t = 20$ ,  $r = 4\%$ ,  $k = 1\,000$ .

## Výpočet progresivního splácení

Uvažujme stejně jako u anuitního splácení zapůjčenou částku  $PV$  a úrokovou sazbu pro jedno úrokové období  $r$ . Současně předpokládejme, že dluh bude splacen přesně po  $t$  splátkách  $a_i$ , kde  $a_i$  odpovídá splátce v  $i$ -tém úrokovém období. Potom musí platit rovnice:

$$\left( \left( (PV * (r + 1) - a_1) * (r + 1) - a_2 \right) * \dots \right) * r - a_t = 0 \quad (2.20)$$

Dále víme, že výše každé další splátky narůstá o konstantu  $k$  oproti splátce předchozí, tudíž platí:

$$a_{i+1} = a_i + k \quad (2.21)$$

Obecněji lze napsat:

$$a_i = a_1 + k * (i - 1) \quad i \in 1..t \quad (2.22)$$

Pokud dosadíme do rovnice 2.20 vyjádření každé proměnné  $a_i$  pomocí obecného vzorce z rovnice 2.22, členy rovnice roznásobíme a vytkneme neznámé  $a_1$  a  $k$  získáme rovnici:

$$PV * (1 + r)^t = a_1 * \left( \sum_{i=0}^{t-1} (1 + r)^i \right) + k * \left( \sum_{i=0}^{t-2} (t - i - 1) * (1 + r)^i \right) \quad (2.23)$$

Z rovnice 2.23 lze odvodit vztahy pro výpočet výše libovolné splátky  $a_i$  pro  $i$  v rozsahu 1..t (rovnice 2.24), výši konstanty  $k$  o kterou narůstají jednotlivé splátky (rovnice 2.25) a původní půjčenou částku  $PV$  (rovnice 2.26):

$$a_i = \frac{PV * (r + 1)^t - k * \left( \sum_{j=0}^{t-2} (t - j - 1) * (1 + r)^j \right)}{\sum_{j=0}^{t-1} (1 + r)^j} + (i - 1) * k \quad (2.24)$$

$$k = \frac{PV * (1 + r)^t - a_1 * \left( \sum_{i=0}^{t-1} (1 + r)^i \right)}{\sum_{i=0}^{t-2} (t - i - 1) * (1 + r)^i} \quad (2.25)$$

$$PV = \frac{a_1 * \left( \sum_{i=0}^{t-1} (1 + r)^i \right) + k * \left( \sum_{i=0}^{t-2} (t - i - 1) * (1 + r)^i \right)}{(1 + r)^t} \quad (2.26)$$

Opět je ovšem nutné brát v potaz, že rovnice 2.24, 2.25 a 2.26 uvažují, že úroková sazba je pro každé úrokové období stejná. Pokud by se úroková sazba  $r_i$  pro různá  $i$ -tá úroková období mohla lišit, je nutné počítat s následujícími vztahy:



$$a_i = \frac{PV * \left( \prod_{j=1}^t (1 + r_j) \right) - k * \left( t - 1 + \sum_{j=3}^t \left( (j - 2) * \prod_{k=j}^t (1 + r_k) \right) \right)}{1 + \sum_{j=2}^t \prod_{k=j}^t (r_k + 1)} + k * (i - 1) \quad (2.27)$$

$$k = \frac{PV * \left( \prod_{j=1}^t (1 + r_j) \right) - a_1 * \left( 1 + \sum_{j=2}^t \prod_{k=j}^t (r_k + 1) \right)}{(t - 1) + \sum_{j=3}^t \left( (j - 2) * \prod_{k=j}^t (1 + r_k) \right)} \quad (2.28)$$

$$PV = \frac{a_1 * \left( 1 + \sum_{j=2}^t \prod_{k=j}^t (r_k + 1) \right) + k * \left( (t - 1) + \sum_{j=3}^t \left( (j - 2) * \prod_{k=j}^t (1 + r_k) \right) \right)}{\prod_{j=1}^t (1 + r_j)} \quad (2.29)$$

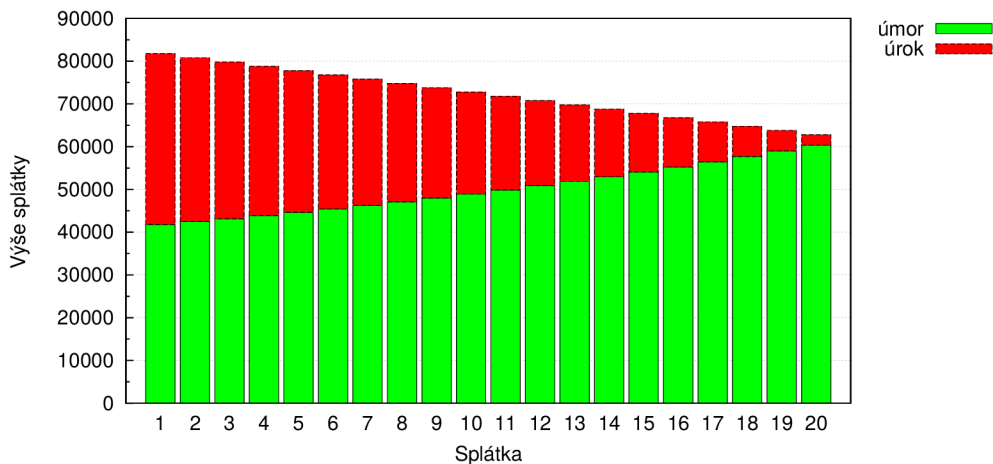
### 2.7.3 Degresivní splácení

Přesným opakem progresivního splácení je degresivní splácení, kdy na počátku úvěru je výše splátky nejvyšší a v každém úrokovém období je o předem stanovenou konstantu snižována.

Tuto metodu nejčastěji vyžadují lidé starší, kteří jsou v aktuální době dobře zajištěni, mohou si dovolit v současnosti splácet více a dosáhnout tak minimalizování finanční zátěže směrem ke stáří.

Lze říct, že veškeré výhody progresivního splácení jsou nevýhodou degresivního a naopak. Vzhledem k vysokým splátkám na počátku splácení úvěru dochází výrazně rychleji k umořování jistiny, a tudíž je minimalizována úroková část. Na druhou stranu si tuto metodu splácení nemůže dovolit každý.

Průběh degresivního splácení lze vidět na grafu 2.4.



Obrázek 2.4: Vývoj anuity a poměru úrokové a úmorové části půjčky v čase u degresivního splácení.  $PV = 1\,000\,000$ ,  $t = 20$ ,  $r = 4\%$ ,  $k = -1\,000$ .

## Výpočet degresivního splácení

Pro výpočet degresivního splácení lze použít naprosto stejné vzorce jako u splácení progresivního (tj. rovnice 2.27, 2.28 a 2.29). Rozdíl je poté pouze ve znaménku konstanty  $k$ , která nabývá kladných hodnot v případě progresivního a záporných v případě degresivního splácení.

## 2.8 Splátkový kalendář

Splácení úvěru probíhá dle tzv. splátkového kalendáře (někdy označovaného jako umořovací plán). Lze si jej představit jako tabulku (obrázek 2.5), jejíž každý řádek odpovídá jedné splátce a sloupce nesou informace o pořadí, výši a datu splátky, výši úrokové a úmorové části splátky a o zůstatku dluhu. Současně zde mohou být uvedeny i informace o poplatcích za vedení půjčky, pokud si nějaké společnost účtuje.

Order	Due date	Annuity	Interest	Amortization	Charges	Balance
1. annuity	23.3.2015	121 200.00	50 510.00	62 690.00	8 000.00	937 310.00
2. annuity	22.4.2015	121 200.00	47 340.00	65 860.00	8 000.00	871 450.00
3. annuity	22.5.2015	121 200.00	44 020.00	69 180.00	8 000.00	802 270.00
4. annuity	21.6.2015	121 200.00	40 520.00	72 680.00	8 000.00	729 590.00
5. annuity	21.7.2015	121 200.00	36 850.00	76 350.00	8 000.00	653 240.00
6. annuity	20.8.2015	121 200.00	32 990.00	80 210.00	8 000.00	573 030.00
7. annuity	19.9.2015	121 200.00	28 940.00	84 260.00	8 000.00	488 770.00
8. annuity	19.10.2015	121 200.00	24 690.00	88 510.00	8 000.00	400 260.00
9. annuity	18.11.2015	121 200.00	20 220.00	92 980.00	8 000.00	307 280.00
10. annuity	18.12.2015	121 200.00	15 520.00	97 680.00	8 000.00	209 600.00
11. annuity	17.1.2016	121 200.00	10 570.00	102 630.00	8 000.00	106 970.00
12. annuity	16.2.2016	120 400.00	5 430.00	106 970.00	8 000.00	0.00
<b>Total paid</b>		<b>Total interest</b>		<b>Total amortization</b>		<b>Total charges</b>
1 453 600.00		357 600.00		1 000 000.00		96 000.00

Obrázek 2.5: Splátkový kalendář vygenerovaný vyvíjenou aplikací. Půjčka = 1 000 000, roční úroková sazba = 60%, fixní poplatky = 8 000, úrokových období = 12.

Lze se také setkat s pojmy anuitní, progresivní nebo degresivní splátkový kalendář, tato označení ovšem nijak nemění metadata týkající se struktury splátkového kalendáře, pouze definují, jakou metodou splácení jsou jednotlivé položky vypočítány.

Jelikož je anuitní splácení v současnosti nejběžnější metodou a společnost HCI jiné metody nepoužívá, budeme dále pracovat pouze s pojmem splátkový kalendář, a to vždy ve smyslu anuitní splátkový kalendář.

## Kapitola 3

# Návrh simulátoru splátkového kalendáře

Hlavním účelem výsledné aplikace je poskytnutí jednoduše ovladatelného, intuitivního, snadno rozšiřitelného a modifikovatelného simulátoru splátkového kalendáře, pro účely testování a konfigurování úvěrových balíčků způsobem, který odpovídá metodám použitých společností HCI pro generování splátkového kalendáře pro zákazníky v Bělorusku.

Jelikož je aplikace vyvíjena pro mezinárodní společnost, je velice vhodné, aby veškeré texty, které aplikace poskytuje, byly psány anglicky.

Vedle těchto základních požadovaných vlastností simulátoru je také nutné vytvořit aplikaci multiplatformní, co nejméně závislou na dalších technologiích, a to tak, aby generovala splátkový kalendář v reálném čase.

Tato kapitola pojednává o výběru použitých technologií (viz 3.1), požadovaných vstupech, které je aplikaci nutno předat (viz 3.2), negativních vlivech zaokrouhlování, které byly v průběhu návrhu odhaleny (viz 3.3) a způsobu výstupu vygenerovaného splátkového kalendáře (viz 3.4) tak, aby byly splněny všechny požadavky na výslednou aplikaci.

### 3.1 Použité technologie

Za účelem co nejmenší technologické závislosti je aplikace vytvořena v programovacím jazyce Javascript s použitím knihovny `jQuery` (více viz kapitola 4.1). Vizualizace je realizována značkovacím jazykem `HTML` s použitím kaskádových stylů.

Vybraná kombinace technologií vyžaduje pro používání aplikace pouze některý z internetových prohlížečů (viz 5.1) a je tedy i dobře přenositelná a relativně nezávislá.

### 3.2 Vstupy simulátoru

Tato podkapitola pojednává o jednotlivých simulátorem vyžadovaných vstupních hodnotách z hlediska jejich významu, povinnosti zadání, formátu, definičnímu oboru a jejich vlivu na výsledný splátkový kalendář.

Veškeré vstupy jsou získány vyplněním a odesláním `HTML` formuláře, viz obrázek 3.1.

Loan type	SS <input checked="" type="radio"/> SC <input type="radio"/> MS <input type="radio"/>	Date of signature	<input type="text"/>
Loan amount	<input type="text"/>	First installment date	<input type="text"/>
Interest rate	<input type="text"/>	Decimal point rounding	<input type="text" value="8"/>
Installments	<input type="text"/>	Annuity rounding	<input type="text" value="10"/>
Annuity proc	<input type="text"/>	int. and amor. rounding	<input type="text" value="10"/>
Monthly fee	<input type="text" value="0"/>		
<input type="button" value="Create schedule"/>			

Obrázek 3.1: Příklad vstupního HTML formuláře z vyvíjené aplikace.

### 3.2.1 Výše úvěru (Loan amount)

Výše úvěru je povinně zadaná hodnota reprezentující původní zapůjčenou částku, tzv. jistinu (viz 2.3). Jelikož jsou obvykle v Bělorusku poskytovány k vypůjčení relativně vysoké finanční částky (souvisí s mírou inflace v této destinaci) je definiční obor hodnot zadané jistiny omezen na pouze přirozená čísla v rozsahu 1 000 - 1 000 000 000 (tento rozsah byl zvolen společností HCI).

### 3.2.2 Úroková sazba (Interest rate)

Úroková sazba (viz 2.2) je povinnou položkou a reprezentuje roční úrokovou sazbu (p.a.) zadanou v procentech (symbol % se nezadává). Definičním oborem jsou tedy všechna reálná čísla v intervalu  $< 0; \infty >$ . Jelikož se však úročí na základě denní úrokové sazby (p.d.), je nutné roční úrokovou sazbu převést na denní.

Způsob výpočtu denní úrokové sazby reprezentuje následující rovnice 3.1:

$$r_d = \frac{r_a}{d} \quad (3.1)$$

Kde  $r_d$  je denní úroková sazba,  $r_a$  je roční úroková sazba a  $d$  je počet dní v daném roce.

Z rovnice 3.1 lze vidět, že výše denní úrokové sazby, na základě které je počítána výše úrokové části anuity, je závislá na tom, zda je či není přestupný rok.

### 3.2.3 Datum podepsání smlouvy a první splátky

Datum podepsání smlouvy (*date of signature*) a datum první splátky (*first installment date*) jsou povinné hodnoty a pro jejich zadání je použita komponenta frameworku JQuery UI s názvem `Datepicker` (více viz kapitola 4.1), která uživateli nabídne možnost vybrat konkrétní datum v zobrazeném kalendáři. Data jsou zobrazována evropským způsobem ve formátu *dd-mm-rrrr*.

Datum podepsání smlouvy určuje počátek počítání úroků. V tento den z legislativních důvodů nesmí být zahájeno úročení, a proto úročení začíná následující den.

Datum první splátky poté závisí na datu podepsání smlouvy a typu půjčky (popsáno níže 3.2.4) a určuje datum, kdy musí být uhrazena 1. anuita.

Pro snadnější zadávání umožňuje aplikace v případě půjček typu SS a SC automaticky dopočítat datum první splátky po zadání data podpisu a naopak (u půjček typu MS automatické dopočítání není možné).

### 3.2.4 Typ půjčky (Loan type)

Běloruští klienti, pro které EmbedIT vyvíjí IT řešení, poskytují tři typy úvěrů:

- hotovostní (anglicky *cash loan*), značeno SC,
- spotřebitelský (anglicky *consumer loan*), značeno SS,
- úvěr s měsíčním splácením, značen MS.

Hotovostní úvěr znamená, že věřitel poskytne jistinu bez nutnosti udání účelu, naopak spotřebitelská půjčka je poskytnutí jistiny za účelem koupě konkrétního produktu.

Z implementačního hlediska spočívá rozdíl mezi SS a SC úvěry de facto pouze v denní prodlevě mezi podepsáním smlouvy o poskytnutí půjčky a datem první splátky (viz 3.2.3). V případě hotovostního úvěru je první splátka hrazena po 30 dnech od podepsání smlouvy a u spotřebitelského úvěru už po 29. Všechny ostatní anuity jsou hrazeny po 30 dnech.

Důsledkem o den kratšího úrokového období před první splátkou (v případě spotřebitelského úvěru) je snížení úrokové části první a tím pádem i následujících splátek, jelikož je aplikováno složené úročení v rámci každého úrokového období.

Vztah pro výpočet procentuální části úroku v rámci jedné splátky a vliv délky úrokového období na výši úroku lze vidět v následující rovnici 3.2:

$$p = \left(1 + \frac{r_d}{100}\right)^d - 1 \quad (3.2)$$

Kde  $p$  je výše úroku vyjádřena jako desetinné číslo (tj. 1% = 0.01),  $r_d$  je denní úroková sazba v procentech (viz rovnice 3.1) a  $d$  je počet dní úrokového období.

V případě, že je úrokové období na přelomu přestupného a nepřestupného roku, je nutné výši úroku počítat upravenou rovnicí 3.3, která vychází z rovnice 3.2:

$$p = \left(1 + \frac{r_{d1}}{100}\right)^{d_1} * \left(1 + \frac{r_{d2}}{100}\right)^{d-d_1} - 1 \quad (3.3)$$

Kde  $d_1$  je počet dní od počátku úrokového období do 1. ledna,  $r_{d1}$  je výše denní úrokové sazby (p.d.) v roce na počátku úrokového období a  $r_{d2}$  je výše denní úrokové sazby na konci úrokového období ( $r_{d1}$  a  $r_{d2}$  lze vypočítat dosazením do rovnice 3.1).

#### Měsíční splácení

Dalším typem splácení, které bylo v Bělorusku nově nasazeno, je zavedení nového způsobu výpočtu SS a SC úvěrů, využívající měsíčního splácení (značeno MS). Tento princip je pro zákazníky přívětivější, jelikož spočívá ve hrazení splátek vždy ke konkrétnímu předem stanovenému datu. Přívětivost spočívá v tom, že si zákazník nemusí složitě dopočítávat, kolikátý den v měsíci má splátku uhradit jako u SS a SC půjček, kdy datum další splátky je vždy 30 (s výjimkou první splátky u SS půjčky) dní po splátce předchozí, ale například vždy 15. dne aktuálního měsíce.

Princip počítání úroku zůstává stejný jako u SS a SC půjček, tedy založený na rovnicích 3.2 a 3.3 s tím rozdílem, že počet dní  $d$  není vždy 30, ale závisí na konkrétním měsíci, ve kterém má být splátka uhrazena.

Současně je zavedeno pravidlo, že splátka nesmí být hrazena k 28.-31. dni v měsíci, jelikož některé měsíce tolik dní nemají (výjimku tvoří 28. den měsíce, 28 dní má sice každý měsíc, ale i tak je z interních důvodů vyloučen).

Podepsáním smlouvy o půjčce (dle parametru popsaném v kapitole 3.2.3) začíná 1. úrokovací období (počítání úroků začíná až den následující), přičemž 1. splátka musí být uhrazena za 15-45 dní od podepsání smlouvy v domluvený n-tý den měsíce splňující výše zmíněná kritéria.

### 3.2.5 Počet splátek (installments)

Je povinně zadaná hodnota, určující počet splátek, které bude muset dlužník uhradit. Definičním oborem jsou tedy nenulová přirozená čísla. Běloruští zákazníci dále omezují počet splátek na rozsah minimálně 1 a maximálně 99 anuit, mezi kterými je třicetidenní rozestup v případě SS a SC úvěrů. V případě měsíčního splácení se počet dní mezi jednotlivými splátkami liší (popsáno v kapitole 3.2.4).

### 3.2.6 Fixní poplatek (Monthly fee)

Fixní poplatek představuje částku konstantní výše, kterou je nutné v rámci každé anuity uhradit. Jedná se o povinný vstup, jehož implicitní hodnota je nastavena na 0. Z interních důvodů běloruských zákazníků je tento vstup definován jako celočíselná hodnota v rozmezí 0 – 50 000.

### 3.2.7 Výše anuity (Annuity proc)

Výše anuity reprezentuje procentuální část z původní zapůjčené jistiny, kterou je nutné na konci každého úrokového období uhradit. Jelikož se jedná o procentuální hodnotu, její rozsah je omezen na reálná čísla v intervalu  $< 0; 100 >$  (symbol % se neuvádí).

Procentuální hodnota anuity není povinným vstupem a nastavením její výše na 0, nebo nenastavením této hodnoty dojde k jejímu automatickému dopočítání. Automatické odvození výše anuity je nezbytné z důvodu testování a jednoduché konfigurace úvěrových balíčků.

V případě, že je procentuální anuita zadána koncovým uživatelem, může dojít k vygenerování nevalidního splátkového kalendáře, tj. dluh může být splacen dříve než po zadaném počtu splátek, nebo naopak nemusí být celý dluh umořen. V případě generování splátkového kalendáře s dopočítáním anuity dojde k umoření dluhu přesně po zadaném počtu splátek.

### Způsob automatického výpočtu anuity

Pro odvození rovnice automatického výpočtu anuity je nutné nejprve odvodit vztah pro výpočet aktuální dlužné částky po jednom úrokovém období. Po každém úrokovém období vzroste dlužná částka o úroky (viz rovnice 3.2 a 3.3) spolu s poplatky a poté je snížena o hodnotu anuity. Výpočet dlužné částky po jednom úrokovém období ukazuje rovnice 3.4:

$$l_{i+1} = l_i * (p_{i+1} + 1) + c - a \quad (3.4)$$

Kde  $l_{i+1}$  je výše dluhu po  $i$ -tém úrokovém období,  $l_i$  je výše dluhu před  $i$ -tém úrokovém období,  $p_i$  je výše úroku pro  $i$ -té úrokové období (viz rovnice 3.2 a 3.3),  $c$  je výše poplatků účtovaných měsíčně a  $a$  je výše jedné splátky.

Dále víme, že dluh má být splacen po  $n$  úrokových obdobích, tudíž výše dluhu po  $n$ -tém úrokovém období musí být rovna nule. Rozšíříme tedy rovnici 3.4 na:

$$l_n = \left( \left( \left( l_0 * (p_1 + 1) + c - a \right) * (p_2 + 1) + c - a \right) * \dots \right) * (p_n + 1) + c - a = 0 \quad (3.5)$$

Kde  $l_0$  je původní výše zapůjčeného kapitálu a  $l_n$  je výše dluhu po poslední splátce (ta musí být rovna 0).

Roznásobením členů rovnice 3.5, vytknutím výše anuity ( $a$ ) a poplatku ( $c$ ) získáme vztah:

$$l_n = l_0 * \left( \prod_{i=1}^n (1 + p_i) \right) + (c - a) * \left( 1 + \sum_{i=2}^n \prod_{k=i}^n (1 + p_k) \right) = 0 \quad (3.6)$$

Vztah pro výpočet výše anuity získáme vyjádřením neznámé  $a$  z rovnice 3.6:

$$a = c + \frac{l_0 * \left( \prod_{i=1}^n (1 + p_i) \right)}{1 + \sum_{i=2}^n \prod_{k=i}^n (1 + p_k)} \quad (3.7)$$

Rovnice 3.7 nám poskytuje způsob výpočtu výše anuity  $a$ . Jelikož ovšem aplikace vyžaduje anuitu vyjádřenou jako procentuální hodnotu z původní zapůjčené částky, musíme provést ještě poslední krok výpočtu, dle rovnice 3.8:

$$a_p = \frac{a}{l_0} * 100 \quad (3.8)$$

Kde  $a_p$  je procentuální výše anuity,  $a$  je skutečná výše anuity a  $l_0$  je původně zapůjčená jistina.

### Příklad výpočtu anuity

Uvažujme následující úvěr (informace o měně finančních částek je zanedbána, jelikož nemá vliv na princip výpočtu):

- výše úvěru (*loan amount*):  $l_0 = 1\,000\,000$  (viz 3.2.1),
- typ úvěru (*loan type*): *SS* (viz 3.2.4),
- roční úroková sazba (*interest rate*):  $r_a = 40\%$  (viz 3.2.2),
- výše poplatků za jedno úrokové období (*monthly fee*):  $c = 1\,000$  (viz 3.2.6),
- počet splátek (*installments*):  $n = 4$  (viz 3.2.5),
- datum podepsání smlouvy (*date of signature*): 10. října 2015 (viz 3.2.3),
- datum uhrazení první splátky (*first installment date*): 7. listopadu 2015 (viz 3.2.3),
- pro úplnost je procentuální vyjádření anuity zaokrouhlováno na 3 desetinná místa (viz 3.2.9) a celková anuita je zaokrouhlována na 10 jednotek (viz dále 3.2.8).

Postup výpočtu pro automatické odvození výše anuity je následující:

1. Vypočítáme denní úrokovou sazbu pro přestupný ( $r_{d1}$ ) a nepřestupný ( $r_{d2}$ ) rok využitím rovnice 3.1:

$$r_{d1} = \frac{r_a}{366} = \frac{40}{366} \doteq 0.10929\%$$

$$r_{d2} = \frac{r_a}{365} = \frac{40}{365} \doteq 0.10959\%$$

2. Vypočítáme úrokovou část (vyjádřenou již jako desetinné číslo) pro každé úrokové období využitím rovnic 3.2 a 3.3 (v případě přelomu přestupného a nepřestupného roku).

č.	splatnost	výše úroku	rovnice
1.	07.11.2015	$p_1 = \left(1 + \frac{0.10959}{100}\right)^{29} - 1 \doteq 0.0322735$	<b>3.2</b>
2.	07.12.2015	$p_2 = \left(1 + \frac{0.10959}{100}\right)^{30} - 1 \doteq 0.0334048$	<b>3.2</b>
3.	06.01.2016	$p_3 = \left(1 + \frac{0.10959}{100}\right)^{24} * \left(1 + \frac{0.10929}{100}\right)^6 - 1 \doteq 0.0333862$	<b>3.3</b>
4.	05.02.2016	$p_4 = \left(1 + \frac{0.10929}{100}\right)^{30} - 1 \doteq 0.0333119$	<b>3.2</b>

3. Nyní již známe veškeré proměnné nutné pro dosazení do rovnice **3.7**:

$$a = 1\,000 + \frac{1\,000\,000 * (1 + p_1) * (1 + p_2) * (1 + p_3) * (1 + p_4)}{1 + (1 + p_2) * (1 + p_3) * (1 + p_4) + (1 + p_3) * (1 + p_4) + (1 + p_4)} \doteq 271\,915.85$$

4. Je zvykem přepočítat celkovou anuitu na její procentuální vyjádření z vypůjčené částky, k tomu využijeme rovnici **3.8**:

$$a_p = \frac{a}{l_0} * 100 \doteq \frac{271\,915.85}{1\,000\,000} * 100 \doteq 27.191585\%$$

5. Posledním krokem je správně zaokrouhlit vypočítanou procentuální část a celkovou výši anuity (popis způsobu zaokrouhlování lze nalézt v podkapitolách **3.2.8**, **3.2.9** a **3.3**):

$$a_p \doteq 27.192\% \qquad a \doteq 271\,920$$

Výsledek na závěr porovnáme se splátkovým kalendářem, který je vygenerovaný vyvíjeným simulátorem, a u kterého byla automaticky vypočítána anuita. Na obrázku **3.2** můžeme vidět, že vypočítané anuity jsou si rovny, a že použití takto vypočítané anuity pro tento ukázkový úvěr opravdu vede k umoření dluhu přesně po 4 splátkách. Také si všimněme, že poslední splátka pro tento splátkový kalendář je o 30 jednotek nižší, než splátka předchozí, což je způsobeno vlivem zaokrouhlování výše anuity. Více o problematice zaokrouhlování výše anuity lze nalézt v kapitole **3.3**.



Installment Schedule						
Loan type	Loan amount	Interest Rate	Installments	Annuity proc	Fix fee	
SS	1 000 000,00	40 %	4	27.192 %	1 000,00	
Order	Due date	Annuity	Interest	Amortization	Fees	Balance
1. annuity	7.11. 2015	271 920,00	32 270,00	238 650,00	1 000,00	761 350,00
2. annuity	7.12. 2015	271 920,00	25 430,00	245 490,00	1 000,00	515 860,00
3. annuity	6.1. 2016	271 920,00	17 220,00	253 700,00	1 000,00	262 160,00
4. annuity	5.2. 2016	271 890,00	8 730,00	262 160,00	1 000,00	0,00
Total paid	Total interest	Total amortization	Total fees			
1 087 650,00	83 650,00	1 000 000,00	4 000,00			

Obrázek 3.2: Simulátorem vygenerovaný splátkový kalendář typu *SS* s výší jistiny 1 000 000, výší fixních poplatků 1 000, první splátkou 7.11.2015, vypůjčený na 4 měsíce, s úrokovou sazbou 40 % a se zaokrouhlováním annuity na 3 desetinná místa a 10 jednotek.

### 3.2.8 Zaokrouhlení na počet jednotek

Zaokrouhlení na počet jednotek je povinná celočíselná hodnota v rozmezí 1 - 1 000 000.

Vstupy tohoto typu jsou dva. První určuje, na kolik jednotek se zaokrouhluje výše jedné splátky (*annuity rounding*) a druhý na kolik jednotek se zaokrouhlují informace reprezentující úrokovou a úmorovou část této splátky (*interest and amortization rounding*). Význam těchto atributů je ovšem stejný a určuje, jakou minimální hodnotou musí být všechny vypočítané finanční částky beze zbytku dělitelné.

Důvod, proč tyto hodnoty nejsou zvoleny konstantně spočívá ve vysoké inflaci v Bělorusku, pro které je aplikace vyvíjena (ke dni 7. května 2015 byla naměřena míra inflace 16.21 % viz <http://www.tradingeconomics.com/belarus/inflation-cpi>). Vzhledem k současnému kurzu Běloruského rublu (BYR) je nyní používáno zaokrouhlování na 10 jednotek pro výši jedné annuity a zaokrouhlování na 10 jednotek pro výši úmorové a úrokové části. Tyto hodnoty jsou pro aplikaci vyvíjeného simulátoru implicitní, pokud nejsou explicitně změněny.

### 3.2.9 Zaokrouhlení na desetinná místa (Decimal point rounding)

Hodnota vstupu reprezentující zaokrouhlení na desetinná místa je povinnou celočíselnou hodnotou v rozmezí 0–10. Určuje, na kolik desetinných míst budou zaokrouhovány vstupy zadávané v procentech, tj. výše annuity a roční úroková sazba.

V současnosti je pro běloruské systémy používáno zaokrouhlování těchto hodnot na 8 desetinných míst a z tohoto důvodu je číslo 8 nastaveno jako implicitní.

## 3.3 Vliv zaokrouhlování na generovaný splátkový kalendář

Pokud bychom zaokrouhlovali výši annuity způsobem typickým pro matematiku, tj. pokud je zbytek po zaokrouhlení vyšší nebo roven polovině zaokrouhlovací konstanty zaokrouhlujeme nahoru, jinak dolů, mohlo by dojít k situacím, kdy poslední splátka bude nepatrně vyšší než splátky předchozí. Legislativou je však pro danou destinaci (Bělorusko) vyžadováno, aby poslední splátka byla menší než splátky předchozí. Z tohoto důvodu je nutné výši splátky zaokrouhlovat výhradně nahoru, čímž lze poměrně jednoduše tomuto problému předejít.

Současně vzniká také problém se zaokrouhlováním poslední splátky. Ta je vždy zaokrouhlena nahoru, ovšem pokud by došlo u takto zaokrouhlené splátky k běžnému rozdělení na

úmorovou a úrokovou část, nemusela by splňovat pravidlo, že součet jednotlivých úmorových částí dává dohromady původně zapůjčenou jistinu. Proto tento přebytek po zaokrouhlení u poslední splátky musí být zahrnut pouze do úrokové části.

Další komplikace mohou vzniknout zaokrouhlováním procentuální výše anuity na  $n$  desetinných míst. Skutečná výše anuity (částka, kterou dlužník pravidelně splácí v každém úrokovém období) je totiž závislá na jejím procentuálním vyjádření z původně vypůjčené částky. Problém tudíž nastane, pokud je původně vypůjčená částka nepřiměřeně vysoká k  $n$  desetinným místům, na které se zaokrouhlí procentuální vyjádření anuity. V takovém případě může dojít k nepříjemně vysokému minimálnímu počtu jednotek, který lze procentuálním vyjádřením anuity nastavit. Například pro půjčku ve výši 10 000 000 se zaokrouhlováním na 3 desetinná místa nelze nastavit u skutečné výše anuity řád jednotek ani desítek (tudíž zaokrouhlování výše anuity např. na 10 jednotek částečně ztrácí smysl). Lze tedy sestavit podmínku, kterou by bylo vhodné pro zaokrouhlování dodržovat, aby k těmto nepříjemnostem nedocházelo:

$$L * \frac{1}{10^{dr+2}} \leq ur \quad (3.9)$$

Kde  $L$  je původně vypůjčená částka,  $dr$  je počet desetinných míst, na které se zaokrouhluje a  $ur$  je počet jednotek, na které se zaokrouhluje výše anuity.

Úpravou rovnice 3.9 lze získat vztah pro výpočet vhodné konstanty zaokrouhlování na desetinná místa ( $dr$  zaokrouhleno nahoru):

$$dr = \log_{10} \frac{L}{ur} - 2 \quad (3.10)$$

Pokud nastavíme konstantu pro zaokrouhlování na desetinná místa podle rovnice 3.10, automaticky vypočítáme procentuální výši anuity dle podkapitoly 3.2.7 a z procentuálního vyjádření anuity vypočítanou skutečnou výši anuity zaokrouhlíme na  $ur$  jednotek (zaokrouhlení anuity na počet jednotek), budou všechny splátky vyšší nebo rovny splátce poslední a rozdíl mezi běžnou (tj. jinou než poslední) a poslední splátkou bude minimální.

Společnost HCI však tento vztah v současnosti nepoužívá a tudíž je nutné výslednou výši běžné splátky (ve smyslu jakákoliv kromě poslední) iterativně dopočítat.

### Příklad výpočtu konstanty pro zaokrouhlování na desetinná místa

Uvažujme následující splátku:

- Výše půjčky (*loan amount*):  $L = 2\,500\,000$ ,
- Zaokrouhlování anuity na 10 jednotek (*annuity rounding*):  $u_r = 10$ .

Výpočet vhodného počtu desetinných míst ( $u_r$ ), na které se zaokrouhluje procentuální výše anuity, lze provést dosazením do rovnice 3.10:

$$d_r = \log_{10} \frac{L}{ur} - 2 = \log_{10} \frac{2\,500\,000}{10} - 2 \doteq 3.39794$$

Výsledek je nutné zaokrouhlit nahoru, tudíž vhodný počet desetinných míst procentuální výše anuity pro půjčku ve výši 2 500 000 je 4.

### Výhody přinášející výpočet konstanty zaokrouhlování na desetinná místa

Pokud nevyužijeme vztah 3.10, máme pro nastavení počtu desetinných míst procentuální výše anuity následující možnosti:

- Stanovit konstantní výši počtu desetinných míst, a to na co možná nejvyšší počet, čímž zamezíme nutnosti iterativně vhodnou výši splátky dopočítávat, ovšem za cenu nepřehlednosti v případě uvedení takového čísla na smlouvě.
- Stanovit konstantní výši počtu desetinných míst, a to na číslo nižší, které na smlouvě sice "dobře vypadá", ovšem přináší (především u vyšších vypůjčených částkách) nutnost iterativně dopočítat skutečnou výši anuity.
- Metodou "pokus omyl" zvolit vhodný počet desetinných míst pro každý konkrétní úvěr, což je velice nepraktické.

Výpočet vhodného, resp. dostačujícího počtu desetinných míst pro vyjádření procentuální výše anuity (viz 3.10) přináší dvě následující možnosti pro zlepšení volby počtu desetinných míst:

- Stanovit konstantní výši počtu desetinných míst, ovšem s aplikováním výpočtu 3.10 pro nejvyšší možnou půjčku (v současnosti 1 000 000 000) a aktuálně používaný počet jednotek, na které se skutečná výše anuity zaokrouhluje (v současnosti 10). Dosazením do rovnice 3.10 zjistíme, že v současnosti lze pro libovolný úvěr zaokrouhlovat procentuální výši anuity na 6 desetinných míst.
- Pro každý úvěr určit automatizovaně využitím rovnice 3.10 dostačující počet desetinných míst.

Obě předchozí metody umožňují minimalizovat počet desetinných míst pro vyjádření procentuální výše anuity, aniž by došlo k nutnosti iterativního dopočítávání skutečné výše anuity.

## 3.4 Výstup simulátoru

Výstup simulátoru lze rozdělit do dvou fází. První fáze spočívá v průběžném testování zadaných vstupů, a v případě nevalidní hodnoty je vhodné koncového uživatele (testera) ihned upozornit na nastalou chybu a její příčinu.

Úkolem druhé fáze je vyhodnocení validně vyplněného formuláře vstupů, dopočítání potenciálně neznámé procentuální výše anuity a vygenerování splátkového kalendáře (někdy též umořovacího plánu) jak je popsán v kapitole 2.8.

Aplikací vygenerovaný splátkový kalendář obsahuje informace o:

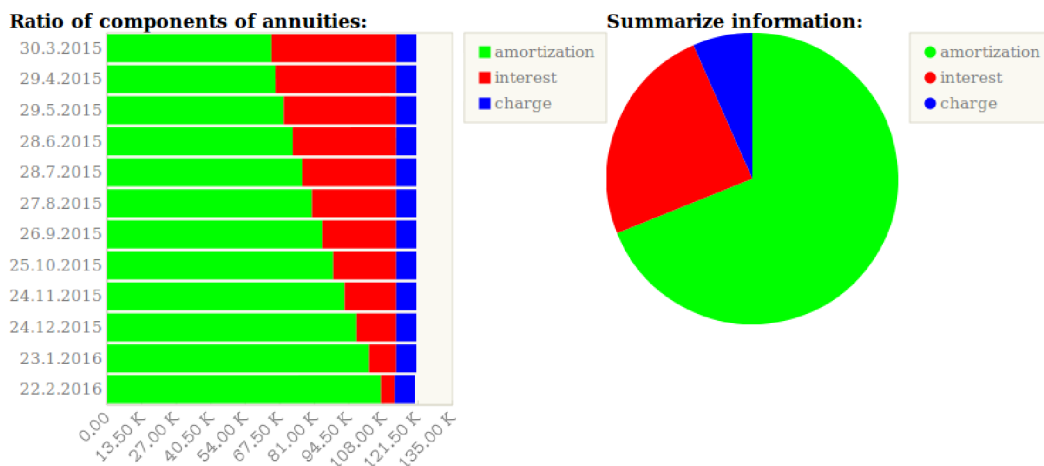
- pořadí splátky (*order*),
- úmorové části anuity (*amortization*),
- datu splatnosti (*due date*),
- výši všech poplatků (*fee*),
- výši splátky (*annuity*),
- úrokové části anuity (*interest*),
- zůstatku dlužné částky (*balance*).

Na konci vytvořené tabulky reprezentující umořovací plán jsou uvedeny sumarizační informace obsahující celkově uhrazenou sumu ručiteli, a jakou část z této sumy tvoří úroky

a poplatky. Pro úplnost sumarizace je zde uvedena i celkově umořená částka, která musí být rovna původnímu zapůjčenému kapitálu (viz obrázek 2.5).

Pro zlepšení vizualizace a usnadnění analýzy informací, které vygenerovaný splátkový kalendář poskytuje, je výstup dále rozšířen o grafické znázornění konkrétního úvěru pomocí grafů.

Informace týkající se jednotlivých anuit a jejich dílčích částí (poplatky, úmorová a úroková část) jsou zahrnuty ve sloupcovém grafu (obrázek 3.3, levý graf) a sumarizační informace týkající se celého úvěru jsou znázorněny koláčovým grafem (obrázek 3.3, pravý graf).



Obrázek 3.3: Ukázka generovaných grafů vyvíjenou aplikací. Půjčka = 1 000 000, splátek = 12, úroková sazba = 60 %, poplatky = 8 000.

## Kapitola 4

# Implementace

Tato kapitola se zabývá podrobným popisem implementace vytvořené aplikace simulátoru splátkového kalendáře. Pojednává podrobněji o veškerých použitých technologiích (podkapitola 4.1), které byly pro vývoj zvoleny, způsobu dosažení cílů (podkapitola 4.2) stanovených v kapitole týkající se návrhu (viz 3) a vysvětlení důležitých implementačních detailů (podkapitola 4.3), jejichž znalost je potřebná pro snadnou rozšiřitelnost a udržitelnost výsledné aplikace.

### 4.1 Popis použitých technologií

Jak je již zmíněno v kapitole zabývající se návrhem (3), funkcionality aplikace je realizována jazykem Javascript verze 1.5. Pro vizualizaci je použit značkovací jazyk HTML verze 5 s využitím kaskádových stylů.

Pro zlepšení interakce mezi jazykem Javascript a HTML využívá aplikace knihovnu JQuery verze 2.1.1 s MIT licencí [5]. JQuery je současně nezbytnou součástí pro možnost používat widget DatePicker [6] (jenž je součástí plug-inu jQuery UI vyvíjeného společně s knihovnou jQuery pod MIT licencí [7]), YUI charts verze 3.17.2 s BSD licencí [8] a jQuery Validation plugin verze 1.13.1 [9].

DatePicker frameworku JQuery UI slouží jako usnadnění pro uživatele při zadávání data, kdy je uživateli umožněn výběr ze zobrazeného kalendáře a není tak nutné zadávat datum v textovém formátu. Možnost zadávat datum i textovou formou je ovšem zachována a používá evropský formát (*dd-mm-rrrr*).

YUI (The Yahoo! User Interface Library) charts poskytuje aplikaci simulátoru splátkového kalendáře prostředky pro automatizovanou grafickou vizualizaci v podobě sloupcových a koláčových grafů, které jsou použity pro zlepšení srozumitelnosti výstupních informací.

jQuery Validation plugin umožňuje definovat pomocí struktury ve formátu JSON<sup>1</sup> specifikaci jednotlivých vstupních informací a zajistit tak automatizovanou kontrolu typu a rozsahu zadaných dat.

---

<sup>1</sup>JSON: JavaScript Object Notation (JavaScriptový objektový zápis). Na platformě nezávislý datový formát, více na [10].

## 4.2 Dosažení stanovených cílů

Jedním z požadavků, popsaným v kapitole zabýbající se návrhem (3), je vytvořit multiplatformní a především na ostatních technologiích co nejméně závislou aplikaci. Tento cíl je velmi snadno splnitelný vhodným zvolením použitých technologií (viz 4.1), kdy pro správný běh aplikace stačí pouze moderní webový prohlížeč (viz 5.4). V případě uložení implementace knihovny jQuery, widgetu DatePicker, YUI charts a jQuery Validation plugin na pevný disk společně s implementací simulátoru splátkového kalendáře, je aplikace i schopna běžet off-line bez nutnosti připojení k internetu.

Dalším z cílů je dosáhnout intuitivní a snadno ovladatelné aplikace. Jelikož má aplikace sloužit pro testovací a konfigurační účely úvěrů ve společnosti HCI, netřeba aplikaci zatěžovat složitou grafikou jako jsou animace a podobné nadbytečnosti, které by mohli naopak negativně ovlivnit přehlednost výstupních informací nebo znesnadnit snadnou realizaci vstupu.

Naopak je velice důležité, aby aplikace umožňovala co možná nejefektivněji zadávat vstupní informace, především protože jich aplikace vyžaduje poměrně mnoho. Vstupy, které jsou velice časté (například zaokrouhlovací konstanty - viz 3.2.8 a 3.2.9) je vhodné implementovat jako výchozí. V případě špatně zadaného vstupu je dobré ihned uživatele informovat o invaliditě vstupu spolu se stručnou nápovědou, nikoli s oznámením čekat až na odeslání požadavku pro vygenerování výstupu, jelikož takový přístup vede ke zpomalení procesu zadávání vstupních informací (konkrétní ukázkou takového ošetření vstupů lze vidět na obrázku 4.1). Po zažádání o vygenerování výstupu je současně velice vhodné ve vstupním formuláři uchovat zadané informace, jelikož v případě konfiguračního využití je typické generovat výstup opakovaně se změnou obvykle pouze malého množství vstupních informací.

Uchování vstupních informací i po aktualizaci stránky webového prohlížeče, ve kterém je aplikace spuštěna, je zajištěno uložením těchto informací za tzv. kotvu (anglicky nejčastěji *anchors*), jenž se nachází za symbolem # v URL adrese této stránky.

Loan amount	<input type="text" value="1000000"/>
Interest rate	<input type="text" value="-1"/>
	Please enter a value greater than or equal to 0.
Installments	<input type="text" value="12"/>

Obrázek 4.1: Ukázka způsobu validace vstupů aplikací simulátoru splátkového kalendáře.

Dalším podstatným požadavkem na aplikaci je informačně bohatý výstup, který uživateli poskytne všechny informace nutné ke snadnému testování a konfiguraci úvěrových balíčků. Pro zlepšení vizualizace výstupu je vedle tabulkového splátkového kalendáře vygenerována i jeho grafická reprezentace (popsáno v kapitole 3.4).

## 4.3 Implementační detaily

Tato podkapitola má za úkol seznámit čtenáře s podrobnějším popisem implementace dílčích částí, ze kterých aplikace sestává a poskytnout tak i informace užitečné pro její snadnou udržitelnost a rozšiřitelnost.

### 4.3.1 Vstupní informace

Veškeré vstupní informace jsou zadávány prostřednictvím HTML formuláře dostupného v `installment_schedule.html` (otevření tohoto souboru ve webovém prohlížeči vede ke spuštění aplikace simulátoru).

Jednotlivé vstupní položky jsou poté propojeny s `JQuery Validation pluginem`, a tudíž v případě přidání nového vstupního atributu stačí současně přidat novou položku do struktury `rules` dostupnou v modulu `files/jquery/validator/validator_settings.js` a nastavit její volitelné parametry `required`, `digits` a `range` pro automatické nastavení validace vstupu.

Veškeré vstupní parametry jsou po potvrzení formuláře (stisknutím tlačítka `create schedule`) předány objektu `Inputs` implementovaném v modulu `files/inputs.js` prostřednictvím kterého se vstupy distribuují dalším objektům.

### 4.3.2 Generování splátkového kalendáře

Objekt `Inputs` popsáný výše (4.3.1) je vstupem konstruktoru objektu `Scheduler`, který je implementovaný modulem `files/scheduler.js`, jenž obstarává veškerou funkcionalitu týkající se výpočtu splátkového kalendáře.

Výpočet je realizován implementací matematických vztahů z rovnic 3.3 a 3.7. Metodou `CalculateInterests` získáme pole jednotlivých úrokových částí pro každé úrokové období. Vypočítané úrokové sazby jsou nezbytné pro automatizované odvození anuity (metoda `CalculateAnnuity`).

Pokud již známe procentuální výši anuity a procentuální vyjádření úrokových částí pro všechna úroková období, lze vypočítat splátkový kalendář metodou `CalculateSchedule`, který je posléze uložen v maticovém tvaru, kde každý řádek odpovídá jedné anuitě včetně všech jí příslušejících informací.

### 4.3.3 Generování výstupních informací

Pro vizualizaci splátkového kalendáře uloženého jako atribut objektu `Scheduler` slouží moduly `files/tables.js`, který z maticového tvaru vytvoří HTML tabulku a `files/charts/charts.js`, který tytéž informace vizualizuje graficky pomocí koláčového grafu pro sumarizační informace a sloupcového grafu pro informace o jednotlivých splátkách. Ukázky vizualizace lze vidět na obrázcích 2.5 a 3.3.

Pro grafickou vizualizaci byla použita knihovna `YUI charts` s BSD licencí, která umožňuje pomocí `JSON` struktury přesně definovat vzhled a datový obsah grafů. Popis definice generovaných grafů lze nalézt v modulech `files/charts/annuity_chart.js` a `files/charts/summarize_chart.js`.

### 4.3.4 Podpůrné moduly

V podkapitolách 4.3.1, 4.3.2 a 4.3.3 jsou popsány stěžejní moduly realizující funkcionalitu aplikace simulátoru splátkového kalendáře. Tyto moduly poskytují hlavní výpočetní rozhraní, které je kompletně řízeno modulem `files/behavior.js`, který reaguje na žádost o vytvoření splátkového kalendáře (stiskem tlačítka `create schedule`).

Některé hlavní moduly používají podpůrné funkce poskytované podpůrnými moduly. Mezi tyto moduly patří:

- `files/autocontrols.js` pro automatické doplňování vstupních informací (např. závislost data podpisu a data první splátky viz [3.2.3](#)),
- `files/number_ops.js` poskytující rozšíření některých číselných operací (např. rozšířené zaokrouhlování na libovolný počet jednotek, formátování číselných informací pro finanční reprezentaci),
- `files/date_ops.js` pro operace s daty (např. posun o  $N$  dní, počet dní do konce roku, počet dní mezi 2 daty apod.).



## Kapitola 5

# Testování

Hlavním cílem testování simulátoru splátkového kalendáře je ověřit, že veškeré jím vygenerované výsledky jsou plně identické s hodnotami získanými z oficiálního generátoru splátkového kalendáře nasazeného v Bělorusku, vytvořeného společností HCI. Tato vlastnost je naprosto nezbytná pro plnění účelu aplikace, jímž je usnadnění testování, ověřování dat generovaných oficiálním splátkovým kalendářem a provádění experimentů pro určení vhodné konfigurace úvěrových balíčků.

Dalším důležitým aspektem je uživatelské testování, spočívající ve zpětné vazbě od testerů společnosti HCI, pro které je aplikace primárně určena (více v podkapitole 5.3).

### 5.1 Průběh testování funkcionality

Ověřování funkcionality aplikace probíhalo průběžně, vždy po dokončení implementace některého logického celku. Za první funkční logický celek bylo považováno pouhé jádro aplikace, umožňující vytvářet základní SS a SC půjčky. Po otestování jádra aplikace se postupně přidávala různá rozšíření splátkového kalendáře (korektní řešení fixních poplatků, zaokrouhlování apod.) a k nim probíhalo příslušné ověřování validity výsledků.

Pro ověřování validity výsledků byl k dispozici interní nástroj společnosti HCI, umožňující na základě specifikace vstupů pro konkrétní úvěr automatizovaně vygenerovat skutečný splátkový kalendář a poskytnout jej ve formátu HTML tabulky. Výsledné tabulky reprezentující splátkové kalendáře z vyvíjené a oficiální aplikace byly porovnávány, v případě shody prohlášeny za funkční a v případě neshody následovala podrobná analýza rozdílů výsledků a případně jejich konzultace.

### 5.2 Problémy a úskalí při testování

Jelikož je simulovaný generátor splátkového kalendáře aktuálně nasazen v Bělorusku, oblast úvěrů je závislá na legislativě a společnosti poskytující úvěry se snaží neustále zdokonalovat služby pro své klienty, dochází k častým změnám a úpravám, na které je nutné brát zřetel a veškeré změny reflektovat i pro simulátor.

Tento problém je částečně řešen co možná největší parametrizací vstupů simulátoru a snahy o snadno rozšiřitelný kód, aby v případě změny bylo její zanesení do simulátoru co nejjednodušší. I tak je ovšem v případě změny nutné opakovaně testovat již jednou otestovanou část.

## 5.3 Uživatelské testování

Vzhledem k relativně malé koncové skupině uživatelů (testerské a vývojářské oddělení dedikovaného týmu společnosti HCI pro Bělorusko) proběhlo uživatelské testování formou demonstrační prezentace, která měla za úkol seznámit budoucí uživatele s aplikací a se způsobem, jak s ní pracovat. Po nasazení testovací aplikace následovalo získání zpětné vazby a oprava nevhodně navržených prvků.

Většina nedostatků spočívala spíše v detailech aplikace, tudíž nedošlo k nutnosti kompletně měnit její funkcionalitu a návrh. Mezi odhalené a opravené nedostatky patří:

- Nahrazení desetinné tečky desetinnou čárkou pro usnadnění přenosu vygenerované tabulky do programu Microsoft Excel.
- Přidání hlavičky obsahující zadané vstupní parametry do vygenerované tabulky (opět z důvodu možnosti přenosu do Microsoft Excel).
- Změna fungování aplikace v případě manuálního zadání nevhodné výše procentuální anuity. Aplikace původně rozpoznávala, že anuita byla zadána špatně a končila informací o chybě bez generování výstupu. V současnosti dochází ke generování výstupu i pro nevalidně zadanou výši anuity.
- Nahrazení pojmu *fix charges* (fixní poplatky) za *fix fees*.

Obrázek 5.1 Ukazuje vzhled vygenerovaného splátkového kalendáře s reflexí požadavků koncových uživatelů aplikace.

Improperly chosen annuity proc!

Installment Schedule						
Loan type	Loan amount	Interest Rate	Installments	Annuity proc	Fix fee	
SS	1 000 000,00	10 %	5	66.000 %	5 000,00	
Order	Due date	Annuity	Interest	Amortization	Fees	Balance
1. annuity	8.5. 2015	660 000,00	7 980,00	647 020,00	5 000,00	352 980,00
2. annuity	7.6. 2015	360 890,00	2 910,00	352 980,00	5 000,00	0,00
Total paid	Total interest	Total amortization	Total fees			
1 020 890,00	10 890,00	1 000 000,00	10 000,00			

Obrázek 5.1: Splátkový kalendář vygenerovaný s nevhodně zvolenou anuitou, včetně hlavičky se vstupními parametry a s použitím desetinných čárek.

## 5.4 Podpora webovými prohlížeči

Jelikož je vyvíjená aplikace psána v jazyce JavaScript a používá řadu dalších knihoven a plug-inů, bylo nutné provést kontrolu podpory webovými prohlížeči.

Tabulka 5.1 ukazuje seznam otestovaných a podporovaných operačních systémů a webových prohlížečů. V případě spuštění aplikace v některém z operačních systémů či webových prohlížečů, jenž v tabulce 5.1 nejsou uvedeny, může způsobit problémy s vizualizací či podporou některých plug-inů. Na základě informací týkajících se podpory webovými prohlížeči, uvedených na oficiálních stránkách použitých knihoven a pluginů (viz [5, 6, 7, 8, 9]), lze ovšem předpokládat dobrou podporu většiny moderních webových prohlížečů (tj. poslední či předposlední stabilní verze prohlížečů jako jsou Internet Explorer, Google Chrome, Mozilla Firefox, Opera).

Operační systém	Mozilla Firefox	Google Chrome	Opera	Internet Explorer
Windows XP	20 - 37	37 - 42	25 - 29	nepodporován
Windows 7	20 - 37	36 - 42	23 - 29	10 - 11
Windows 8	20 - 37	36 - 42	23 - 29	10 - 11
Windows 10	32 - 37	37 - 42	23 - 29	11
Linux (Ubuntu)	37	42	29	—

Tabulka 5.1: Tabulka testovaných a podporovaných operačních systémů a webových prohlížečů.

## 5.5 Výsledky testování simulátoru

Aplikace simulátoru splátkového kalendáře byla finálně testována porovnáním s oficiálním generátorem splátkového kalendáře, používaném ke dni 26.4. 2015 v Bělorusku.

Závěrečné komplexní testování probíhalo vygenerováním různých splátkových kalendářů pro všechny typy úvěrů (*SS*, *SC* a *MS* – viz kapitola 3.2.4) jak oficiální, tak vyvíjenou aplikací. Současně bylo snahou testovat skutečně přípustné splátkové kalendáře, tj. takové, které odpovídají některým z poskytovaných úvěrových balíčků. Tyto potenciálně možné úvěry byly dále voleny tak, aby zahrnovaly pokud možno co největší spektrum různých specifik (fixní poplatky, přelom přestupného a nepřestupného roku, apod.). Vygenerované výsledky byly porovnány a v případě neshody byly analyzovány rozdíly.

Výjimku tvoří testy řešící problematiku vlivu zaokrouhlovacích konstant, jelikož v oficiálním generátoru splátkového kalendáře jsou pevně nastaveny a nelze je snadno měnit. Proto jejich testování probíhalo pouze využitím simulátoru a detailní analýzy. Při tomto testování nebyly odhaleny žádné problémy.

Závěrečné komplexní testování odhalilo chybu u simulování půjček typu *MS* (viz 3.2.4). Zde v některých případech došlo k špatnému výpočtu počtu dní jednoho úrokového období, a proto nebyly vygenerované splátkové kalendáře ve shodě. Po odstranění tohoto nedostatku již poskytoval oficiální generátor i simulátor stejné výsledky.

Dalo by se tedy říct, že oficiální generátor i simulátor splátkového kalendáře jsou ke dni 26.4. 2015 výpočetně ekvivalentní, ovšem je nutné upozornit, že generátor splátkového kalendáře společnosti HCI je stále vyvíjen, tudíž tato ekvivalence může být snadno porušena a pro její zachování je nutné simulátor neustále udržovat.

### 5.5.1 Ukázka použitých testů a jejich výsledků

Následující tabulky 5.2, 5.3 a 5.4 poskytují ukázky některých použitých testů a jejich výsledků. Všimněme si, že výsledky poskytované simulátorem (levé části tabulek) jsou na-prosto shodné s výsledky poskytovanými oficiálním generátorem (pravé části tabulek).

Všechny tři ukázkové testy (5.2, 5.3 a 5.4) využívají stejné zaokrouhlovací konstanty, které jsou ke dni 26.4. 2015 v Bělorusku používány:

- Zaokrouhlování anuity a úrokové sazby na 8 desetinných míst,
- zaokrouhlování anuity na 10 jednotek,
- zaokrouhlování úrokové a úmorové části na 10 jednotek.

## Test půjčky typu *SC*

Vstupní parametry:

- výše půjčky (*loan amount*): 10 000 000,
- roční úroková sazba (*interest rate*): 39,9 %,
- počet splátek (*installments*): 12,
- výše poplatků za 1 úrokovací období (*Monthly fee*): 5 000,
- datum podpisu (*date of signature*): 1.1. 2015,
- datum první splátky (*first installment date*): 31.1. 2015.

Procentuální výše anuity byla automaticky odvozena simulátorem splátkového kalendáře s výsledkem 10,29632095, z čehož plyne skutečná výše anuity  $10\,000\,000 * 0,1029632095 = 1\,029\,632,095$ , tj. zaokrouhloveno nahoru 1 029 640.

č.	Vyvíjený simulátor						Oficiální generátor					
	datum	anuita	úrok	úmor	poplatek	zůstatek	datum	anuita	úrok	úmor	poplatek	zůstatek
1.	31.1.2015	1 029 640	333 200	691 440	5 000	9 308 560	31.1.2015	1 029 640	333 200	691 440	5 000	9 308 560
2.	2.3.2015	1 029 640	310 160	714 480	5 000	8 594 080	2.3.2015	1 029 640	310 160	714 480	5 000	8 594 080
3.	1.4.2015	1 029 640	286 350	738 290	5 000	7 855 790	1.4.2015	1 029 640	286 350	738 290	5 000	7 855 790
4.	1.5.2015	1 029 640	261 750	762 890	5 000	7 092 900	1.5.2015	1 029 640	261 750	762 890	5 000	7 092 900
5.	31.5.2015	1 029 640	236 330	788 310	5 000	6 304 590	31.5.2015	1 029 640	236 330	788 310	5 000	6 304 590
6.	30.6.2015	1 029 640	210 070	814 570	5 000	5 490 020	30.6.2015	1 029 640	210 070	814 570	5 000	5 490 020
7.	30.7.2015	1 029 640	182 930	841 710	5 000	4 648 310	30.7.2015	1 029 640	182 930	841 710	5 000	4 648 310
8.	29.8.2015	1 029 640	154 880	869 760	5 000	3 778 550	29.8.2015	1 029 640	154 880	869 760	5 000	3 778 550
9.	28.9.2015	1 029 640	125 900	898 740	5 000	2 879 810	28.9.2015	1 029 640	125 900	898 740	5 000	2 879 810
10.	28.10.2015	1 029 640	95 950	928 690	5 000	1 951 120	28.10.2015	1 029 640	95 950	928 690	5 000	1 951 120
11.	27.11.2015	1 029 640	65 010	959 630	5 000	991 490	27.11.2015	1 029 640	65 010	959 630	5 000	991 490
12.	27.12.2015	1 029 530	33 040	991 490	5 000	0	27.12.2015	1 029 530	33 040	991 490	5 000	0

Tabulka 5.2: Test porovnání simulátoru a oficiálního generátoru splátkového kalendáře u *SC* typu úvěru.

## Test půjčky typu *SS*

Vstupní parametry:

- výše půjčky (*loan amount*): 2 000 000,
- roční úroková sazba (*interest rate*): 47 %,
- počet splátek (*installments*): 12,
- výše poplatků za 1 úrokovací období (*Monthly fee*): 3 000,
- datum podpisu (*date of signature*): 1.1. 2015,
- datum první splátky (*first installment date*): 30.1. 2015.

Procentuální výše anuity byla automaticky odvozena simulátorem splátkového kalendáře s výsledkem 10,75204477, z čehož plyne skutečná výše anuity  $2\,000\,000 * 0,1075204477 = 215\,040,8954$ , tj. zaokrouhleno nahoru 215 050.

č.	Vyvíjený simulátor						Oficiální generátor					
	datum	anuita	úrok	úmor	poplatek	zůstatek	datum	anuita	úrok	úmor	poplatek	zůstatek
1.	30.1.2015	215 050	76 050	136 000	3 000	1 864 000	30.1.2015	215 050	76 050	136 000	3 000	1 864 000
2.	1.3.2015	215 050	73 370	138 680	3 000	1 725 320	1.3.2015	215 050	73 370	138 680	3 000	1 725 320
3.	31.3.2015	215 050	67 910	144 140	3 000	1 581 180	31.3.2015	215 050	67 910	144 140	3 000	1 581 180
4.	30.4.2015	215 050	62 240	149 810	3 000	1 431 370	30.4.2015	215 050	62 240	149 810	3 000	1 431 370
5.	30.5.2015	215 050	56 340	155 710	3 000	1 275 660	30.5.2015	215 050	56 340	155 710	3 000	1 275 660
6.	29.6.2015	215 050	50 210	161 840	3 000	1 113 820	29.6.2015	215 050	50 210	161 840	3 000	1 113 820
7.	29.7.2015	215 050	43 840	168 210	3 000	945 610	29.7.2015	215 050	43 840	168 210	3 000	945 610
8.	28.8.2015	215 050	37 220	174 830	3 000	770 780	28.8.2015	215 050	37 220	174 830	3 000	770 780
9.	27.9.2015	215 050	30 340	181 710	3 000	589 070	27.9.2015	215 050	30 340	181 710	3 000	589 070
10.	27.10.2015	215 050	23 190	188 860	3 000	400 210	27.10.2015	215 050	23 190	188 860	3 000	400 210
11.	26.11.2015	215 050	15 750	196 300	3 000	203 910	26.11.2015	215 050	15 750	196 300	3 000	203 910
12.	26.12.2015	214 940	8 030	203 910	3 000	0	26.12.2015	214 940	8 030	203 910	3 000	0

Tabulka 5.3: Test porovnání simulátoru a oficiálního generátoru splátkového kalendáře u *SS* typu úvěru.

## Test půjčky typu *MS*

Vstupní parametry:

- výše půjčky (*loan amount*): 50 000 000,
- roční úroková sazba (*interest rate*): 55 %,
- počet splátek (*installments*): 12,
- výše poplatků za 1 úrokovací období (*Monthly fee*): 10 000,
- datum podpisu (*date of signature*): 1.5. 2015,
- datum první splátky (*first installment date*): 1.6. 2015.

Procentuální výše anuity byla automaticky odvozena simulátorem splátkového kalendáře s výsledkem 11,12092985, z čehož plyne skutečná výše anuity  $50\,000\,000 * 0,1112092985 = 5\,560\,464,925$ , tj. zaokrouhleno nahoru 5 560 470.

č.	Vyvíjený simulátor						Oficiální generátor					
	datum	anuita	úrok	úmor	poplatek	zůstatek	datum	anuita	úrok	úmor	poplatek	zůstatek
1.	1.6.2015	5 560 470	2 389 180	3 161 290	10 000	46 838 710	1.6.2015	5 560 470	2 389 180	3 161 290	10 000	46 838 710
2.	1.7.2015	5 560 470	2 164 290	3 386 180	10 000	43 452 530	1.7.2015	5 560 470	2 164 290	3 386 180	10 000	43 452 530
3.	1.8.2015	5 560 470	2 076 320	3 474 150	10 000	39 978 380	1.8.2015	5 560 470	2 076 320	3 474 150	10 000	39 978 380
4.	1.9.2015	5 560 470	1 910 310	3 640 160	10 000	36 338 220	1.9.2015	5 560 470	1 910 310	3 640 160	10 000	36 338 220
5.	1.10.2015	5 560 470	1 679 090	3 871 380	10 000	32 466 840	1.10.2015	5 560 470	1 679 090	3 871 380	10 000	32 466 840
6.	1.11.2015	5 560 470	1 551 390	3 999 080	10 000	28 467 760	1.11.2015	5 560 470	1 551 390	3 999 080	10 000	28 467 760
7.	1.12.2015	5 560 470	1 315 420	4 235 050	10 000	24 232 710	1.12.2015	5 560 470	1 315 420	4 235 050	10 000	24 232 710
8.	1.1.2016	5 560 470	1 157 820	4 392 650	10 000	19 840 060	1.1.2016	5 560 470	1 157 820	4 392 650	10 000	19 840 060
9.	1.2.2016	5 560 470	945 380	4 605 090	10 000	15 234 970	1.2.2016	5 560 470	945 380	4 605 090	10 000	15 234 970
10.	1.3.2016	5 560 470	678 090	4 872 380	10 000	10 362 590	1.3.2016	5 560 470	678 090	4 872 380	10 000	10 362 590
11.	1.4.2016	5 560 470	493 780	5 056 690	10 000	5 305 900	1.4.2016	5 560 470	493 780	5 056 690	10 000	5 305 900
12.	1.5.2016	5 560 390	244 490	5 305 900	10 000	0	1.5.2016	5 560 390	244 490	5 305 900	10 000	0

Tabulka 5.4: Test porovnání simulátoru a oficiálního generátoru splátkového kalendáře u *MS* typu úvěru.

# Kapitola 6

## Závěr

Výsledkem práce je komplexní simulátor splátkového kalendáře, jehož výsledky jsou totožné s generátorem splátkového kalendáře nasazeného běloruskými zákazníky společnosti HCI. Simulátor poskytuje širokou škálu vstupních parametrů, díky kterým je umožněno efektivní testování současného generátoru splátkového kalendáře a snadné konfigurování nových úvěrových balíčků.

Celý simulátor je vytvořen s velkým ohledem na obecnost a udržovatelnost, tudíž i v případě zásadnější změny referenčního systému je relativně snadno přizpůsobitelný novému systému. V případě zavedení zásadnější změny je ovšem již nezbytná úprava zdrojového kódu.

Simulátor je multiplatformní, na ostatních technologiích nezávislý (pro správné fungování postačuje některý z moderních webových prohlížečů 5.1) a fungující v off-line režimu. Současně je snadno provozitelný, jelikož nevyžaduje žádnou instalaci.

Celý vývoj jádra simulátoru (matematických principů výpočtu splátkového kalendáře) probíhal bez analýzy již existujících zdrojových kódů generátoru splátkového kalendáře a byl postaven na důkladném prostudování principů finanční matematiky zabývající se úročením a splácením. Tento přístup přinesl odvození obecných vztahů uplatňujících se v této problematice:

- Univerzální rovnice pro popis anuitního, progresivního i degresivního splácení 2.27. Tato rovnice umožňuje v závislosti na výši konstanty  $k$  popisovat anuitní ( $k = 0$ ), progresivní ( $k > 0$ ) a degresivní ( $k < 0$ ) splácení. Současně rovnice zahrnuje i skutečnost, že úroková sazba nemusí být pro všechna úroková období stejná. Rovnice 2.28 a 2.29 poskytují vyjádření ostatních proměnných vystupujících v rovnici 2.27.
- Rovnice 3.7 popisující vztah pro automatické odvození výše anuity ze zadaných vstupních parametrů úvěru. V případě splnění podmínek z podkapitoly 3.3 lze celý tento proces automatického odvození řešit čistě analyticky bez nutnosti iterativních metod.
- Rovnice 3.3 zobecňující problematiku výpočtu úrokové části jedné splátky. Rovnice postihuje i úroková období na přelomu přestupného a nepřestupného roku.

Výše zmíněné, odvozené rovnice poskytují společnosti HCI pevný matematický základ umožňující čistě analytické řešení (bez nutnosti iterativního dopočítávání) generování splátkového kalendáře (minimálně pro běloruské zákazníky) a staly se potenciální možností pro vylepšení současně nasazeného systému.

## 6.1 Možná rozšíření

Jelikož společnost HCI je značně rozsáhlá, nebylo možné zahrnout do simulátoru všechna specifika, která obsahuje současný generátor splátkového kalendáře pro běloruské zákazníky, natož univerzální prostředek pro testování a konfigurování všech typů splátkových kalendářů, které poskytuje společnost HCI.

Proto je prvním, potenciálně možným rozšířením přidat do simulátoru všechny další možné varianty a specifika, které se v současnosti u poskytování úvěrů používají. Jedná se například o benefity pro poctivě splácející klienty jako je odložení úroků, nebo naopak možnosti jako odložení splátek (buďto formou restrukturalizace dluhu zvýšením počtu splátek, nebo odložením splátek na stanovený počet měsíců) pro klienty, kteří se dostanou do finanční tísně a nejsou schopni úvěr splácet.

Další možností je zautomatizování simulátoru a jeho propojení s oficiálně nasazeným generátorem splátkového kalendáře. Vznikly by tak dvě vzájemně propojené ovšem separátně vyvíjené aplikace, přičemž jedna by automatizovaně testovala a v případě problémů reportovala druhou.

Velké možnosti lze najít i v teoretické sféře této problematiky. Velice zajímavé výsledky by mohla projevovat důkladná matematická analýza dalších specifik, které se v oblasti poskytování úvěrů uplatňují a získání tak univerzálního vztahu (například rozšířením rovnice 2.27) pro výpočet libovolného splátkového kalendáře. Takový vztah by poté dokázal simulovat jakýkoliv typ úvěru pouze dosazením všech parametrů do jedné obecné rovnice a vznik nového úvěrového produktu by tak neznamenal nový vývoj, ale pouze stanovení všech těchto parametrů. Další výhodu takového přístupu lze vidět ve zjednodušení testování, jelikož by vše mělo pevný matematický základ a analytické řešení bez nutnosti iterativních metod pro dopočítávání některých parametrů.



# Literatura

- [1] Grobařová, S.: *Základy finanční matematiky*. Tribun EU s.r.o., 2012, ISBN 978-80-263-0253-7.
- [2] Šoba, O.; Širůček, M.; Ptáček, R.: *Finanční matematika v praxi*. Grada Publishing, a.s., 2013, ISBN 978-80-247-4636-4.
- [3] Macháček, O.: *Finanční a pojistná matematika*. Prospektrum, 1995, ISBN 80-7175-035-2.
- [4] Radová, J.; Dvořák, P.; Málek, J.: *Finanční matematika pro každého*. Grada Publishing, a.s., 2013, ISBN 978-80-247-4831-3.
- [5] jQuery [online]. <http://www.jquery.com>, 2015 [cit. 13-3-2015].
- [6] DatePicker [online]. <http://www.jqueryui.com/datepicker>, 2015 [cit. 13-3-2015].
- [7] jQuery user interface [online]. <http://www.jqueryui.com>, 2015 [cit. 13-3-2015].
- [8] The Yahoo! User Interface Library – charts [online]. <http://www.yuilibrary.com/yui/docs/charts>, 2015 [cit. 14-03-2015].
- [9] jQuery Validation Plugin [online]. <http://www.jqueryvalidation.org>, 2015 [cit. 14-03-2015].
- [10] The JSON Data Interchange Format [online]. <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>, říjen 2013 [cit. 14-03-2015].

# Příloha A

## Obsah CD

- `technicka_zprava_src`  
Adresář se zdrojovými soubory technické zprávy.
- `technicka_zprava.pdf`  
Technická zpráva ve formátu PDF.
- `src`  
Adresář se zdrojovými soubory vytvořené aplikace simulátoru splátkového kalendáře.
- `src/installment_schedule.html`  
Soubor, jehož otevřením ve webovém prohlížeči dojde ke spuštění aplikace.