

**FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO**

NÁVRH

Program na výpočet umiestnenia pretekárov v rebríčku

Zimný semester 2018/2019

Monika Vlčková
Tomáš Takács
Adam Trizna
Jozef Mizerík

Obsah

1. Úvod	4
1.1. Účel dokumentu	4
2. Špecifikácia vonkajších interfejsov	4
2.1. Pridanie vstupného súboru	4
2.2. Generovanie výstupného súboru	4
3. Dátové modely	4
3.1. Model perzistentných údajov	4
3.2. Model formátov súborov	4
3.2.1. Formát vstupného súboru	4
3.2.2. Formát konfiguračného súboru	5
3.2.3. Formát výstupného(vygenerovaného) súboru	5
4. Návrh používateľského rozhrania	7
4.1. Návrh interakcie programu cez konzolu s používateľom	7
4.1.1. Možnosti výberu operácie	7
4.1.2. Interakcia s používateľom pri pridávaní vstupného súboru	7
4.1.3. Interakcia s používateľom pri generovaní výstupného súboru	7
5. Rozdelenie na moduly a ich interfejsy	7
5.1. Spúšťačiaci modul	7
5.1.1. Možnosti spustenia	7
5.2. Interakčný modul	8
5.2.1. Spracovanie dát	8
5.2.2. Výstupné dáta	8
5.2.3. Kontrola Vstupov	8
5.3. Modul na prácu so vstupnými súbormi	8
5.3.1. Vstupné dáta	8
5.3.2. Rozdelenie modulu na triedy	9
5.3.3. Pridanie vstupného súboru	9
5.3.4. Načítanie všetkých vstupných súborov	9
5.3.5. Výstupné dáta	10
5.4. Modul na kontrolu vstupného súboru	10
5.4.1. Vstupné údaje	10
5.4.2. Kontrola údajov	10
5.4.3. Výstupné údaje	10
5.5. Konfiguračný modul	10
5.5.1. Vstupné dáta	10

5.5.2. Spracovanie dát	10
5.5.3. Výstupné dáta	11
5.6. Výpočtový modul	11
5.6.1. Vstupné dáta	11
5.6.2. Spracovanie dát	11
5.6.3. Výstupné dáta	11
5.7. Priradovací modul	11
5.7.1. Vstupné dáta	11
5.7.2. Spracovanie dát	11
5.7.3. Výstupné dáta	12
5.8. Generujúci modul	12
5.8.1. Vstupné dáta	12
5.8.2. Spracovanie dát	12
5.8.3. Výstupné dáta	12
6. Návrh implementácie	12
6.1. Use-case diagram	12
6.2. UML component diagram	12
6.3. UML class diagram	12
6.4. UML sequence diagram	12
6.5. Architektúra systému	13
6.6. Použité technológie	13
6.6.1. Java, verzia 9	13
6.6.2. Java Architecture for XML Binding	13
6.6.3. Apache FreeMarker™	13

1. Úvod

1.1. Účel dokumentu

Tento dokument predstavuje návrh architektúry programu na výpočet umiestnenia pretekárov v rebríčku. Dokument je určený pre vývojárov informačného systému. Dokument popisuje funkcie a metódy informačného systému a podáva návrh na implementáciu.

2. Špecifikácia vonkajších interfejsov

2.1. Pridanie vstupného súboru

Aplikácia bude čítať vstupný súbor na používateľom zadanej adrese a súbory uložené medzi perzistentnými údajmi programu.

2.2. Generovanie výstupného súboru

Aplikácia bude generovať html výstupný súbor do priečinku Out, ktorý sa nachádza v priečinku spracovávanej sezóny. Spolu s vygenerovaným hml súborom uloží do tohto priečinku aj dva css súbory pre tlač a pre web.

3. Dátové modely

3.1. Model perzistentných údajov

Stromová štruktúra podpriečinkov priečinka „Data“ (v zložke so zdrojovými súborami) a súborov:

- Data
 - 2018
 - In
 - kolo1.xml
 - kolo2.xml
 - kolo3.xml
 - ...
 - Out
 - vystup1.html
 - vystup2.html
 - vystup3.html
 - ...

3.2. Model formátov súborov

3.2.1. Formát vstupného súboru

Vstupný súbor má rozšírenie .xml.

Vstupné súbory, ktoré bude program vedieť spracovať sú exportované výsledky etapy pretekov zo stránky v tvare “<https://is.orienteeering.sk/competitions/id/export-results>“, kde *id* označuje identifikátor kola pretekov.

Program bude pracovať so vstupnými súbormi, ktorých typ exportu je IOF XML v3.

Koreňovým elementom vstupného súboru je element ResultList.

Štruktúra vstupného súboru:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" iofVersion="3.0" status="Complete"
creator="ISSZOS" xmlns="http://www.orienteering.org/datastandard/3.0">
  <Event>
    <Id type="ISSZOS"></Id>
    <Name></Name>
    <StartTime>
      <Date></Date>
    </StartTime>
  </Event>
  <ClassResult>
    <Class>
      <Id type="ISSZOS"></Id>
      <Name></Name>
      <ShortName></ShortName>
    </Class>
    <PersonResult>
      <Person>
        <Id type="ISSZOS"></Id>
        <Name>
          <Family></Family>
          <Given></Given>
        </Name>
      </Person>
      <Result>
        <Time></Time>
        <Position></Position>
        <Status></Status>
      </Result>
    </PersonResult>
  </ClassResult>
</ResultList>
```

3.2.2. Formát konfiguračného súboru

Konfiguračný súbor je vo formáte *properties*. Názov súboru je *config.properties*. Obsahuje päť premenných, ktoré si používateľ môže zmeniť podľa svojej potreby. Pred každou premennou je komentár popisujúci, čo daná premenná nastavuje.

3.2.3. Formát výstupného(vygenerovaného) súboru

Výstupný súbor je vo formáte html. V hlavičke súboru sa nachádzajú informácie o súťaži, typ bodovania, z koľkých kôl sa započítavajú body do výsledku a kontakt na osobu, zodpovednú za bodovanie. V tele súboru sa nachádza navigácia v ktorej sú jednotlivé kategórie a tabuľka pre každú kategóriu. V tabuľke sa nachádzajú stĺpce pre jednotlivé kolá, poradie súťažiacich, ich mená, priezviská, id a súčet bodov.

Štruktúra výstupného súboru:

```

<html>
  <head>
    <meta charset="utf-8">
    <title>Sanasport Cup</title>
    <link href="stylesheet.css" rel="stylesheet" media="all">
    <link href="styleprint.css" rel="stylesheet" media="print">
  </head>
  <body>
    <header>
      <div>
        <section class = "v1c">
          <b>Sanasport Cup 2018<br>Západná oblasť SZOŠ</b>
        </section>
        
      </div>
      <div>
        <br>Bodovanie: 50-45-40-35-30-29-28-27-...3-2-1-1-1-1... DISK = 0.
        <br>Počíta sa 10 najlepších výsledkov
      </div>
      <div>
        <br>Informácie a reklamácie: <i>Richard Balogh SKS6901</i>
        &lt;<b>balogh@elf.stuba.sk</b>&gt;
      </div>
    </header>
    <nav>
      <a href="#M-10">M-10</a>
    </nav>

    <table>
      <tr>
        <td></td> <td></td> <td></td> <td></td>
        <td></td> <td>1.kolo</td> <td></td>
      </tr>
      <tr>
        <td></td> <td></td> <td></td> <td></td>
        <td></td> <td>06.01</td> <td></td>
      </tr>
      <tr>
        <td></td>
        <td class = "vertical">Zmena oproti predošlému kolu</td>
        <td></td> <td></td> <td></td>
        <td class = "vertical">Trojkráľové preteky (TJ Sokol Pezinok)</td>
        <td>Spolu</td>
      </tr>
      <tr id = "#M-10" class="category">
        <td></td> <td></td>
        <td>M-10</td>
        <td></td> <td></td>
        <td></td> <td></td>
      </tr>
      <tr>
        <td>1</td> <td>=</td>
        <td class = "left">Daniel</td>
        <td class = "left">Ondovcik</td>
        <td>SPE0801</td>
        <td></td>
        <td></td>
      </tr>
    </table>
  </body>
</html>

```

4. Návrh používateľského rozhrania

4.1. Návrh interakcie programu cez konzolu s používateľom

Interakciu s používateľom na konzole zaobstaráva *Interakčný modul*.

4.1.1. Možnosti výberu operácie

Pri spustení programu bude používateľ vyzvaný vybrať si z dvoch ponúknutých možností:

- Možnosť A: Vložiť nový súbor s výsledkami kola, vykonať výpočet bodov (zo všetkých už vložených súborov) a vygenerovať poradie bežcov.
- Možnosť B: Vykonať výpočet bodov a vygenerovať poradie bežcov zo všetkých aktuálnych súborov, bez vkladania nového.

V prípade, že program je spustený z príkazového riadku so zadaným parametrom (viď 5.1.1)

- “-a [adresa_vstupného_súboru]” – vykonáva sa variant možnosti A.
- “-b” – vykonáva sa variant možnosti B.

4.1.2. Interakcia s používateľom pri pridávaní vstupného súboru

V prípade výberu možnosti A (ktorýmkoľvek z možných spôsobov – viď 4.1.1), po zadaní adresy vstupného súboru, *Interakčný modul* adresu spracuje a skontroluje (pre viac informácií o kontrole viď 5.2.3). V prípade negatívneho výsledku kontroly bude používateľ opätovne vyzvaný na zadanie adresy vstupného súboru, v prípade pozitívneho výsledku kontroly *Interakčný modul* pošle adresu *Modulu na prácu so vstupnými súborami*, ktorý s ňou ďalej pracuje.

4.1.3. Interakcia s používateľom pri generovaní výstupného súboru

Po vygenerovaní výsledkov do výstupného .html súboru *Generujúcim Modulom*, bude používateľ informovaný *Interakčným modulom* o dokončení generovania a odkáže ho na výstupný súbor.

Komentár od [AT1]: Tuto je myslím najlepší, keď sa na konzole vypíše len niečo ako „generovanie skončilo, výsledky nájdete vo výstupnom súbore 'vystup.html'“

5. Rozdelenie na moduly a ich interfejsy

5.1. Spúšťací modul

Modul zabezpečuje spustenie programu a naštartovanie kľúčových modulov.

5.1.1. Možnosti spustenia

Program bude možné spustiť z príkazového riadku, alebo spustením .executable súboru. Pri spúšťaní z príkazového riadku bude možnosť zadania jedného z dodatočných parametrov:

- “-a [adresa_vstupného_súboru]”. Zadanie tohto parametra s adresou vstupného súboru znamená, že používateľ si pri spustení automaticky vyberá možnosť A (viď 4.1.1).
- “-b”. Zadanie tohto parametra znamená, že používateľ si pri spustení automaticky vyberá možnosť B (viď 4.1.1).

V prípade nezadania parametra, alebo spustení programu cez .exe súbor dostane používateľ na výber, ktorú z možností A/B si chce vybrať. V prípade výberu možnosti A si program vypýta adresu vstupného súboru na konzole.

Komentár od [TT2]: Myslím, že to môže byť aj .jar súbor

Komentár od [TA3R2]: myslel som to tak, že by to bolo .exe na spustenie .jarka :) .. tak som to robil na iných projektoch, lebo keď dvojklikom spustíš .jarko, zbehne a skončí. Keď dvojklikom spustíš takto vyrobené .exe, spustí sa, zbehne a okno zostane otvorené.

Komentár od [TT4R2]: paráda, dobrá poznámka

5.2. Interakčný modul

Modul zabezpečuje všetkú interakciu s používateľom. Komunikácia prebieha výhradne cez konzolu. Modul taktiež zabezpečuje kontrolu vstupov zadaných na konzole.

5.2.1. Spracovanie dát

5.2.1.1. Vyžiadanie a spracovanie vstupných dát

V prípade, že program tento údaj nedostane už ako parameter volania pri spustení, *interakčný modul* žiada a spracúva vstupné údaje (adresa vstupného súboru), ktoré po kontrole správnosti posúva iným modulom.

5.2.1.2. Vyžiadanie a spracovanie doplňujúcich dát

Niektorý zo zvyšných modulov môže pri spracovaní dát upozorniť na ich neúplnosť a bude potrebné tieto dáta doplniť za behu programu. Vtedy *interakčný modul* používateľovi vysvetlí situáciu, vyžiada zadanie chýbajúcich údajov na konzolu, spracuje ich, skontroluje ich a posunie príslušnému modul, ktorý neúplnosť odhalilo.

5.2.2. Výstupné dáta

Interakčný modul slúži aj na interpretáciu výstupov ostatných modulov používateľovi. Kedykoľvek, keď bude niektorý zo zvyšných modulov chcieť používateľovi predat' nejakú informáciu (napr. upozorniť na nejakú skutočnosť), sformuluje správnu, ktorú posunie *interakčnému modulu* a ten ju zobrazí používateľovi. *Interakčný modul* bude prístupný všetkým ostatným modulom prostredníctvom statických metód.

5.2.3. Kontrola Vstupov

Interakčný modul si pri spracovaní vstupov kontroluje ich správnosť. V prípade negatívneho výsledku kontroly tak môže obratom nanovo vyžiadať od používateľa správne údaje. Kontrola vstupu pozostáva z dvoch častí, kontrola formátu a kontrola relevantnosti údajov.

5.2.3.1. Kontrola formátu

Pri kontrole sa prihliada na formát údajov. Modul zabezpečuje odmietnutie vstupného údajov chýbného formátu.

5.2.3.2. Kontrola relevantnosti

Touto kontrolou sa myslí overenie, či vstupný údaj nesie pravdivú informáciu (pri zadaní adresy súboru sa overí, či sa na zadanej adrese nachádza vôbec nejaký súbor).

5.3. Modul na prácu so vstupnými súbormi

Modul na prácu so vstupnými súbormi zabezpečuje prácu so vstupnými súbormi.

Zabezpečuje načítanie vstupného súboru a zapísanie pracovnej dátovej štruktúry do XML dokumentu.

5.3.1. Vstupné dáta

Pri pridávaní vstupného súboru modul dostane ako argument od *Interakčného modulu* adresu vstupného súboru.

Pri načítavaní všetkých vstupných súborov modul nemá zadávané vstupné dáta.

Komentár od [AT5]: Zatiaľ mám v návrhu len jeden modul na spracovanie súborov („Modul spracujúci vstupný súbor“), ale mali by byť aspoň 2 (Modul parsujúci xml a Modul priradujúci pretekárov) – oboj z týchto modulov môžu žiadať doplnenie dát na konzole.

Komentár od [TT6R5]: Už sú 2 moduly, na chýbajúci údaj upozorňuje „Modul na kontrolu údajov o kole“

Komentár od [AT7]: Lepšie pomenovanie ako relevantnosť mi nenapadlo.

Komentár od [TT8R7]: Myslím, že túto časť môže kľudne riešiť „Modul na prácu so vstupnými súbormi“, ten to aj tak bude kontrolovať automaticky.

Komentár od [AT9R7]: Áno, aj Petrovič na stretnutí spomínal, že by asi bolo vhodnejšie to spojiť do jedného modulu. Aj keď tam sme sa bavili, že sa to spojí s interakčným .. teda s tým, čo zatiaľ spajam.

5.3.2. Rozdelenie modulu na triedy

Modul bude rozdelený na viacero tried, kde každá trieda zabezpečuje inú časť procesu spracovania dát vstupného súboru.

Hlavnou triedou bude trieda zodpovedná za čítanie a zapisovanie Xml súboru.

Okrem hlavnej triedy budú modulu patriť triedy popisujúce elementy Xml dokumentu.

5.3.3. Pridanie vstupného súboru

Modul načíta vstupný súbor na adrese, ktorú dostal ako argument.

Modul vytvorí inštanciu triedy Unmarshaller a zavolá metódu unmarshal, kde ako argument je načítaný súbor.

Vykoná proces unmarshalizácie (*unmarshalling*) a transformuje dáta vstupného súboru do stromovej štruktúry objektov tried. Vytvorí objekt triedy ResultList, ktorá reprezentuje koreňový element vstupného súboru a tvorí tak koreň stromovej štruktúry.

Modul pošle získaný objekt triedy ResultList *Modulu na kontrolu vstupného súboru*.

Modul dostane naspäť od *Modulu na kontrolu vstupného súboru* skontrolovaný a doplnený objekt triedy ResultList.

Modul vyžiada od *Konfiguračného modulu* rok sezóny a uloží ho do premennej seasonYear typu Integer.

Modul vytvorí inštanciu triedy Marshaller a zavolá metódu marshal, kde ako argumenty sú objekt triedy ResultList a adresa nového súboru.

Modul vykoná proces marshalizácie (*marshalling*) a vytvorí z tohto objektu XML dokument.

Modul určí meno dokumentu na základe poradového čísla kola, teda atribútu rank triedy Event, ktorá je súčasťou triedy ResultList. Názov dostane vo forme "kolo{rank}.xml", kde rank zodpovedá hodnote atribútu rank triedy Event.

Modul uloží dokument do priečinka Data, podpriečinka s názvom rovným stringovej reprezentácii hodnoty premennej seasonYear.

5.3.4. Načítanie všetkých vstupných súborov

Modul si pripraví zoznam (objekt triedy List), do ktorého bude postupne pridávať objekty triedy ResultList.

Modul vyžiada od *Konfiguračného modulu* rok sezóny a uloží ho do premennej seasonYear typu Integer.

V priečinku Data prečíta a zapamätá si všetky názvy súborov podpriečinku s názvom rovným stringovej reprezentácii premennej seasonYear.

Modul vytvorí inštanciu triedy Unmarshaller. Modul postupne načíta všetky vstupné súbory na všetkých zapamätaných adresách.

Pre každý vstupný súbor zavolá metódu unmarshal, kde ako argument je načítaný súbor.

Vykoná proces unmarshalizácie (*unmarshalling*) a transformuje dáta vstupného súboru do stromovej štruktúry objektov tried. Vytvorí objekt triedy ResultList, ktorá reprezentuje koreňový element vstupného súboru a tvorí tak koreň stromovej štruktúry.

Modul pošle získaný objekt triedy ResultList *Modulu na kontrolu vstupného súboru*.

Modul dostane naspäť od *Modulu na kontrolu vstupného súboru* skontrolovaný a doplnený objekt triedy ResultList. Tento objekt uloží do pripraveného zoznamu.

5.3.5. Výstupné dáta

Po pridaní vstupného súboru modul uloží vzniknutý XML dokument do úložiska vstupných súborov.

Po načítaní všetkých vstupných súborov modul posielá *Výpočtovému modulu* zoznam so všetkými vytvorenými objektami triedy ResultList ako objekt triedy List.

5.4. Modul na kontrolu vstupného súboru

Modul na kontrolu vstupného súboru zodpovedá za kontrolu a správnosť štruktúry vstupného súboru. Zodpovedá za doplnenie chýbajúcich údajov.

5.4.1. Vstupné údaje

Modul dostane od *Modulu na prácu so vstupnými súbormi* ako argument objekt triedy ResultList.

5.4.2. Kontrola údajov

Modul kontroluje prítomnosť atribútov triedy Event, ktorá je podtriedou triedy ResultList. Sú to nasledovné atribúty:

- *rank* typu Integer (poradové číslo kola)
- *name* typu String (názov kola)
- *startTime* je objekt triedy StartTime, ktorý má atribút *date* typu String. (dátum)

Keď akýkoľvek z týchto atribútov má preddefinovanú hodnotu pre príslušný typ atribútu, tak modul zavolá funkciu *Interakčného modulu* pre príslušný typ atribútu, kde argumentom je názov chýbajúceho atribútu.

Modul dostane od *Interakčného modulu* údaj, ktorý priradí k chýbajúcemu atribútu.

Modul opakuje kontrolu, kým nie sú priradené údaje ku všetkým chýbajúcim atribútom.

5.4.3. Výstupné údaje

Po dokončení kontroly pošle upravený objekt triedy ResultList *Modulu na prácu so vstupnými súbormi*.

5.5. Konfiguračný modul

5.5.1. Vstupné dáta

Program dostane adresu konfiguračného súboru.

5.5.2. Spracovanie dát

Modul prečíta konfiguračný súbor a uloží jeho obsah do dátovej štruktúry (class ConfigurationFile).

5.5.3. Výstupné dáta

Modul poskytne vzniknutý objekt triedy ConfigurationFile ostatným modulom, ktoré závisia od obsahu konfiguračného súboru.

5.6. Výpočtový modul

5.6.1. Vstupné dáta

Modul dostáva dáta od *Modulu na prácu so vstupnými súbormi*. Modul dostane zoznam objektov triedy ResultList, ktorý obsahuje zoznam objektov ClassResult – ide o zoznam kategórií. Objekt ClassResult obsahuje zoznam objektov PersonResult - jedná sa o informácie o jednotlivých pretekároch a ich výsledkoch v danom kole.

Modul dostáva dáta aj od *Konfiguračného modulu* (objekt triedy ConfigurationFile). Tieto dáta obsahujú používateľom zvolený spôsob výpočtu bodov a ďalšie potrebné dáta pre ich výpočet.

5.6.2. Spracovanie dát

Modul prečíta z objektu triedy ConfigurationFile zvolený spôsob bodovania. Na základe toho v triede RoundPointComputation zavolá jednu z troch príslušných funkcií počítajúcich body pretekárov za kolo. Tieto funkcie využívajú aj skupinu súkromných funkcií danej triedy, ktoré vykonávajú čiastkové výpočty. Funkcie na počítanie bodov nastavujú body objektom triedy PersonResult, nič nevracajú.

5.6.3. Výstupné dáta

Objekty triedy ResultList s nastavenou hodnotou bodov za kolo následne spracuje *Priradovací modul*.

5.7. Priradovací modul

5.7.1. Vstupné dáta

Priradovací modul obdrží zoznam objektov triedy ResultList s nastavenými počtami bodov pretekárov za jednotlivé kolá. Počty bodov už predtým vypočíta *Výpočtový modul*.

Tiež využije objekt triedy ConfigurationFile, odkiaľ získa údaj o maximálnom počte kôl, ktoré sa pretekárovi môžu bodovať.

5.7.2. Spracovanie dát

Trieda RunnerAssignment využíva objekty ResultList, ktoré postupne prechádza a priradzuje výsledok k existujúcemu objektu triedy RunnerOverall (ďalej popísanej). Ak takýto objekt neexistuje, vytvorí nový. Priradenie prebieha primárne podľa id, ak pretekár nemá priradené id, priradzuje podľa mena a priezviska. Objekty triedy RunnerOverall sú uložené do slovníka celkových výsledkov, kde kľúč je skratka kategórie a hodnotou je zoznam objektov triedy RunnerOverall.

Trieda RunnerOverall obsahuje zoznam výsledkov pretekára za jednotlivé kolá. Atribút points obsahuje celkový súčet bodov pretekára zohľadňujúci maximálny počet kôl, ktoré sa pretekárovi bodujú. Trieda tiež obsahuje atribút identifier, ktorá primárne obsahuje id pretekára. Ak pretekár nemá id, tvar je “#priezvisko meno”.

5.7.3. Výstupné dáta

Modul odovzdá slovník obsahujúci zoznamy objektov triedy RunnerOverall *Generujúcemu modulu*.

5.8. Generujúci modul

5.8.1. Vstupné dáta

Modul dostáva slovník obsahujúci zoznamy objektov triedy RunnerOverall od *Priradovacieho modulu* a pracuje aj s predpripraveným html šablónou.

5.8.2. Spracovanie dát

Modul spracuje objekty tried a pomocou nástroja Apache FreeMarker a html šablóny vytvorí html súbor.

5.8.3. Výstupné dáta

Výstupom je html súbor s výsledkami pretekov ktorý uloží do priečinku spolu s css súbormi pre tlač a pre web.

6. Návrh implementácie

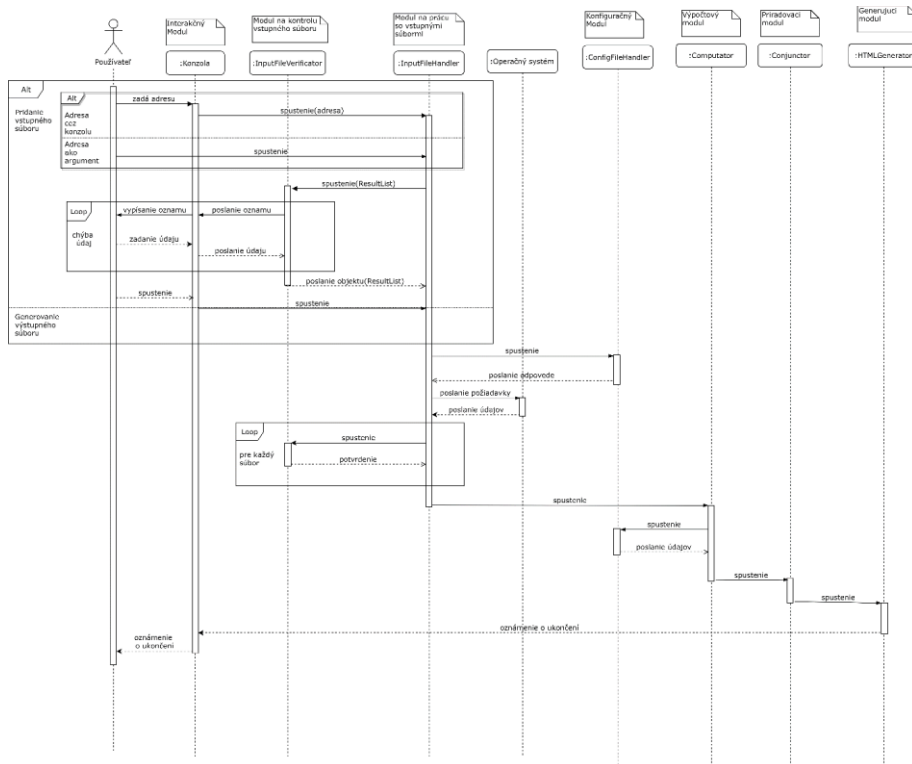
6.1. Use-case diagram



6.2. UML component diagram

6.3. UML class diagram

6.4. UML sequence diagram



6.5. Architektúra systému

6.6. Použité technológie

6.6.1. Java, verzia 9

Programovací jazyk v implementácii programu.

6.6.2. Java Architecture for XML Binding

Skrátene *JAXB*, je knižnica, ktorá sa v programe používa na čítanie a zapisovanie XML dokumentov.

V programe je použitá verzia knižnice *javax.xml.bind:jaxb-api:2.3.1*

6.6.3. Apache FreeMarker™

Je šablónový nástroj, ktorý generuje textový výstup (HTML webové stránky, e-mail, konfiguračné súbory...) založená na šablónach a meniacich sa dátach.

