

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO

TECHNICKÁ DOKUMENTÁCIA

Program na výpočet umiestnenia pretekárov v rebríčku

Zimný semester 2018/2019

Monika Vlčková

Tomáš Takács

Adam Trizna

Jozef Mizerík

Obsah

1 Špecifikácia požiadaviek na softvér	4
1.1 Úvod	4
1.1.1 Predmet špecifikácie	4
1.1.2 Rozsah projektu	4
1.1.3 Slovník pojmov, skratky	4
1.1.4 Odkazy	5
1.2 Všeobecný popis	6
1.2.1 Perspektíva produktu	6
1.2.2 Funkcie produktu	6
1.2.3 Charakteristika používateľov	6
1.2.4 Predpoklady a závislosti	6
1.3 Špecifikácia požiadaviek	7
1.3.1 Štruktúra vstupného súboru *A.....	7
1.3.2 Spustenie programu *A.....	8
1.3.3 Vkladanie výsledkov do úložiska *A.....	8
1.3.4 Spracovanie vstupných súborov *A.....	9
1.3.5 Vyhodnocovanie a bodovanie sezóny.....	10
1.3.6 Vytvorenie dokumentu s priebežným hodnotením z uložených dát	11
1.3.7 Validácia.....	12
1.3.7.1 Nastaviteľná konfigurácia.....	12
1.3.8 Prílohy	13
2 Návrh softvéru	14
2.1 Úvod	14
2.1.1 Účel dokumentu.....	14
2.2 Špecifikácia vonkajších interfejsov.....	14
2.2.1 Pridanie vstupného súboru.....	14
2.2.2 Generovanie výstupného súboru.....	14
2.3 Dátové modely.....	14
2.3.1 Model perzistentných údajov.....	14
2.3.2 Model formátov súborov	15
2.4 Návrh používateľského rozhrania	19
2.4.1 Návrh interakcie programu cez konzolu s používateľom	19
2.5 Rozdelenie na moduly a ich interfejsy	20

2.5.1 Spúšťačiaci modul	20
2.5.2 Interakčný modul	21
2.5.3 Modul na prácu so vstupnými súbormi	22
2.5.4 Modul na kontrolu vstupného súboru	24
2.5.5 Konfiguračný modul	25
2.5.6 Výpočtový modul	25
2.5.7 Prirad'ovací modul	26
2.5.8 Generujúci modul	27
2.6 Návrh implementácie	27
2.6.1 Use-case diagram	27
2.6.2 UML component diagram	28
2.6.3 UML class diagram	28
2.6.4 UML sequence diagram	28
2.6.5 Použité technológie	28
2.6.5.1 Java, verzia 8	28
3 Testovacie scenáre	28
3.1 Testovacie scenáre pre pridávanie vstupného súboru	28
3.2 Testovacie scenáre pre inicializáciu programu	29
3.3 Testovacie scenáre pre interakciu	29

1 Špecifikácia požiadaviek na softvér

1.1 Úvod

1.1.1 Predmet špecifikácie

Táto špecifikácia požiadaviek na softvér popisuje používateľské, funkčné a ostatné požiadavky na program, ktorý slúži na výpočet umiestnenia pretekárov v rebríčku. Dokument je určený pre stakeholderov projektu. Táto špecifikácia požiadaviek je súčasťou dohody medzi zadávateľom a dodávateľom, a bude slúžiť ako východisko pre vyhodnocovanie správnosti fungovania výsledného programu.

1.1.2 Rozsah projektu

Program bude po spustení pracovať na počítači používateľa. Program má za úlohu načítať vstupné súbory s výsledkami kôl, z nich vypočítať umiestnenia pretekárov a vytvoriť výsledný rebríček.

1.1.3 Slovník pojmov, skratky

Dátová štruktúra - spôsob, akým program udržiava dáta v pamäti počítača

Konfiguračný súbor – súbor, v ktorom môže používateľ zmeniť parametre vyhodnocovania.

JRE – Java Runtime Environment.

Kategória – skupina pretekárov zoskupená podľa istých kritérií

Kolo – jeden beh v rámci sezóny. Obvykle sa uskutoční 10 - 14 kôl za rok, ktoré tvoria celú sezónu.

OS - operačný systém

Perzistentné úložisko – miesto v pamäti počítača, na ktoré používateľ nahráva súbory s výsledkami jednotlivých kôl

Pretekár – bežec, ktorý sa zúčastní aspoň jedného kola v rámci sezóny.

Registračný kód - jedinečná postupnosť číslíc a písmen. Každému pretekárovi je pri prvej účasti na behu priradený takýto jedinečný identifikátor, ktorým sa rozlišuje od všetkých zvyšných pretekárov počas celej sezóny.

Sezóna - trvá jeden kalendárny rok a tvoria ju všetky kolá, ktoré sa uskutočnia v tomto časovom období.

Stakeholder - osoba alebo skupina osôb, ktorá sa podieľa na rovnakom projekte.

SZOŠ - Slovenský zväz orientačných športov.

Úložisko dát – priečinok, do ktorého program ukladá kópie vstupných súborov

Vstupný súbor - súbor s výsledkami kola, ktorý používateľ zadá programu na spracovanie

“* A/B/C” - notácia za názvom požiadavky predstavuje jej prioritu. Priority požiadaviek sú rozdelené do troch skupín. Skupina *A – vysoká priorita - požiadavky je nutné splniť pre funkčnosť programu. Skupina *B – stredne vysoká priorita – splnenie požiadavky bude uvítané. Skupina *C – nižšia priorita – splnenie požiadavky predstavuje pridanú hodnotu pre systém.

1.1.4 Odkazy

Propozície rebričkovej súťaže SANASPORT CUP 2018 v orientačnom behu pre rok 2018:

<http://zapad.orienteeing.sk/doc/2018/OR2018.pdf>

Informačný systém SZOŠ:

<https://is.orienteeing.sk/>

1.2 Všeobecný popis

1.2.1 *Perspektíva produktu*

Produktom je konzolová aplikácia, slúžiaca ako nástroj na výpočet umiestnenia pretekárov v rebríčku v sezóne. Aplikácia prideluje pretekárom body podľa spôsobu, ktorý vopred určí používateľ. Produkt dostane vstupné súbory s výsledkami kôl. Program ich spracuje a vypočíta priebežné hodnotenie. Pri každom spustení programu a úspešnom spracovaní vstupného súboru je vygenerovaný dokument so získaným priebežným hodnotením.

1.2.2 *Funkcie produktu*

Hlavnou funkcionalitou produktu bude výpočet umiestnenia pretekárov v rebríčku v sezóne podľa vopred určeného spôsobu vyhodnocovania bodových ziskov. Na požiadanie od používateľa program uloží používateľom zadaný vstupný súbor obsahujúci výsledky pretekárov jedného kola sezóny. Program spracuje všetky uložené súbory s výsledkami pretekárov všetkých kôl sezóny. Po spracovaní údajov pretekárov v súboroch program vytvorí dokument s vypočítaným umiestnením pretekárov v rebríčku v sezóne.

1.2.3 *Charakteristika používateľov*

Používateľom je osoba zodpovedná za spracovanie rebríčka pretekárov.

1.2.4 *Predpoklady a závislosti*

Pre správny chod programu je potrebné, aby súčasťou softvéru počítača, na ktorom bude produkt spúšťaný, bola sada softvérových nástrojov *Java Runtime Environment (JRE)*. Program bude multiplatformový, používať ho bude možné na OS Windows, OS Linux aj MacOS. Vstupný xml súbor bude zadaný korektne.

1.3 Špecifikácia požiadaviek

1.3.1 Štruktúra vstupného súboru

*A

1.3.1.1 Štruktúra vstupného súboru:

1.3.1.1.1 Vo vstupnom súbore sa nachádzajú informácie o kole:

- poradové číslo kola
- názov kola
- dátum uskutočnenia kola

1.3.1.1.2 Program bude akceptovať aj vstupný súbor, v ktorom chýba niektorá z informácií uvedených v bode 3.1.1.1.

1.3.1.1.3 Vo vstupnom súbore sa nachádza zoznam jednotlivých kategórií a ku každej z nich zoznam údajov o pretekároch v danej kategórii

Údaje o pretekárovi:

- kategória v ktorej je pretekár zaradený
- meno pretekára
- priezvisko pretekára
- registračný kód pretekára
- čas zadáný ako počet sekúnd, za ktorý pretekár zabehol dané kolo
- stav, reprezentujúci informáciu o diskvalifikácii
- pozícia, s ktorou pretekár skončil kolo

1.3.1.1.4 V prípade, že bol pretekár diskvalifikovaný, tak jeho čas je 0 sekúnd a údaj označujúci stav označuje dôvod diskvalifikácie

1.3.1.1.5 Ak pretekár nebol diskvalifikovaný, tak stav má hodnotu "OK", a teda pretekárov výsledok sa považuje za platný.

1.3.1.1.6 Program bude akceptovať aj vstup, v ktorom pre niektorých pretekárov nie je zadaná niektorá z kombinácií údajov:

- Ak nie je zadané meno a priezvisko pretekára, tak musí byť zadaný registračný kód pretekára
- Ak nie je zadaný registračný kód pretekára, tak musí byť zadané meno a priezvisko pretekára

Ak nie je zadaný akákoľvek iný údaj pretekára, program si zapamätá:

- názov vstupného súboru, ktorý je načítavaný
- číslo riadku súboru, na ktorom chýba údaj
- názov chýbajúceho údaju

1.3.1.1.7 Program po skončení načítavania súboru vypíše na konzolu všetky zapamätané chýbajúce údaje.

1.3.2 *Spustenie programu* *A

1.3.2.1 Operácie vykonané pri spustení

1.3.2.1.1 Program vypíše používateľovi na konzolu programu pri spustení programu dve možnosti:

Program dá používateľovi možnosť pridať vstupný súbor a uložiť jeho dáta do úložiska dát.

Program dá používateľovi možnosť vytvoriť dokument s priebežným hodnotením z uložených dát.

1.3.3 *Vkladanie výsledkov do úložiska* *A

1.3.3.1 Formát úložiska

Program bude obsahovať priečinok s priečinkami pre jednotlivé sezóny. V týchto priečinkoch sa budú nachádzať xml súbory s výsledkami kôl.

1.3.3.2 Pridanie vstupného súboru a uloženie jeho dát do úložiska dát.

1.3.3.2.1 V prípade výberu možnosti 3.2.1.1.1 pri výbere možností, je používateľ vyzvaný na zadanie cesty k súboru (program rozpozná absolútnu i relatívnu cestu), ktorý chce pridať.

1.3.3.2.2 Vstupný súbor bude načítaný len za predpokladu, že je formátu xml.

1.3.3.2.3 Program prečíta obsah súboru na zadanej adrese a skontroluje, či súbor obsahuje všetky informácie uvedené v bode 3.1.1.1. Ak niektorá z informácií nebude v súbore zadaná, tak v takom prípade program pozastaví načítavanie súboru a vypíše na konzolu názvy všetkých chýbajúcich informácií. Následne program vyzve používateľa tieto údaje zadať do konzoly programu. Program si zadané informácie dočasne uloží, za účelom popísaným v 3.3.2.4.

1.3.3.2.4 Po kontrole obsahu súboru program vytvorí kópiu vstupného súboru. Ak nastalo, že niektoré údaje neboli zadané, program do vzniknutej kópie zapíše zapamätané informácie zadané od používateľa.

1.3.3.2.5 Po vytvorení kópie program načíta z konfiguračného súboru názov sezóny a pokúsi sa nájsť v úložisku dát priečinok so získaným názvom sezóny. Ak takýto priečinok neexistuje, tak ho program vytvorí a následne do tohto priečinku vloží vzniknutú kópiu vstupného súboru.

1.3.3.2.6 Vstupné súbory je možné pridávať v ľubovoľnom poradí.

1.3.3.3 Zmena dát

1.3.3.3.1 V prípade, že používateľ chce zmeniť dáta starších kôl, môže editovať priamo príslušný xml súbor, ktorého sa zmena týka. Tieto zmeny budú zohľadnené pri ďalšej generácii výstupného súboru.

1.3.3.3.2 Používateľ môže súbor v úložisku dát odstrániť a pri ďalšom generovaní výsledkov sa kolo nebude zohľadňovať.

1.3.4 *Spracovanie vstupných súborov*

*A

1.3.4.1 Načítanie vstupných súborov

Program postupne načíta všetky súbory s formátom xml z priečinku zadanej sezóny v úložisku dát.

1.3.4.2 Spracovanie dát zo vstupných súborov

1.3.4.2.1 Pri načítaní vstupných súborov program číta dáta, ktoré obsahujú a ukladá ich do dátových štruktúr v operačnej pamäti, kde sú rozdelené podľa kategórií.

1.3.4.2.2 Program zoradí získané dátové štruktúry z načítaných vstupných súborov podľa poradového čísla kola.

1.3.4.2.3 Program priradí údaje pretekára vo vstupnom súbore k dátam pretekára v dátových štruktúrach len ak sú zaradené do rovnakej kategórie.

1.3.4.2.4 Pre každého pretekára sa program pokúsi priradiť jeho/jej údaje v dátach vstupného súboru k dátam uloženým v dátových štruktúrach. A to nasledovne: Ak je v dátach pretekára zadaný registračný kód, tak program jednoznačne rozhodne, že ide o toho istého pretekára a k pretekárovým dátam v dátovej štruktúre pridá údaj označujúci počet získaných bodov v kole, ktorého výsledky sa spracúvajú.

Ak v dátach pretekára nie je zadaný registračný kód, program sa najprv pokúsi v dátových štruktúrach nájsť všetkých pretekárov s rovnakým menom. Ak sa programu nepodari nájsť takého pretekára, program považuje pretekára za nového a pridá ho ako nový záznam. Program sa posunie do ďalšej fázy výpočtu bodov, len pokiaľ pri spracovaní vstupu nevypísal žiadne chyby.

1.3.5 Vyhodnocovanie a bodovanie sezóny

1.3.5.1 Vyhodnotenie sezóny

*A

Na základe výsledkov jednotlivých kôl je vytváraný celkový rebríček, ktorý obsahuje súhrnné počty bodov pretekárov za všetky, do daného času uskutočnené kolá.

1.3.5.1.1 Do výsledného počtu bodov sa počíta len x najlepších získaných bodových ohodnotení pretekára. Hodnotu x si používateľ nastavuje v konfiguračnom súbore (3.7.1.2).

1.3.5.2 Bodovanie kôl

*A

Predvolené bodovanie jednotlivých kôl je podľa zoznamu v konfiguračnom súbore.

Pretekár na prvom mieste dostáva najvyšší počet bodov, každý ďalší počet bodov rovný nasledujúcej hodnote v zozname. V prípade, že pretekárov je viac ako hodnôt v zozname, všetci nasledujúci pretekári dostávajú počet bodov rovný poslednej hodnote v zozname. V prípade diskvalifikácie je pretekárovi udelených 0 bodov. V prípade

rovnakého umiestnenia viacerých pretekárov sú všetci klasifikovaní na najvyššom nasledujúcom mieste. Umiestnenie pretekárov nasledujúcich po tejto skupine sa o daný počet miest posunie. V prípade, že sa pretekár kola nezúčastnil, nedostane žiadne body (nie nula bodov).

1.3.5.3 Zmena bodovania kôl

Predvolené bodovanie je možné zmeniť na jedno z nasledujúcich:

1.3.5.3.1 Bodovanie podľa počtu účastníkov kola

1.3.5.3.2 Počet účastníkov daného kola je n . Pretekár na prvom mieste obdrží n bodov, postupne každý ďalší o bod menej, až po posledného, ktorý dostane 1 bod. V prípade, že bolo x pretekárov diskvalifikovaných, posledný pretekár, ktorý úspešne dobehne do cieľa dostane $1 + x$ bodov, predposledný $2 + x$ atď.

1.3.5.3.3 Bodovanie podľa dosiahnutého času

1.3.5.3.4 Počet bodov sa odvodí podľa nasledujúceho vzorca:

$$\text{body pretekára } x = \text{čas víťaza} / \text{čas pretekára } x * \text{konštanta}$$

Používateľ si môže konštantu upraviť v konfiguračnom súbore (3.7.1.3).

1.3.6 Vytvorenie dokumentu s priebežným hodnotením z uložených dát

Výstupnou hodnotou programu bude vygenerovaný súbor formátu html. Súbor bude obsahovať výsledky jednotlivých kategórií vo všetkých už uskutočnených kolách zvolenej sezóny. V rámci kategórie budú uvedené výsledky všetkých pretekárov, ktorí sa zúčastnili aspoň jedného z kôl sezóny.

1.3.6.1 Štruktúra vygenerovaného súboru

*A

Výsledky budú pre každú kategóriu obsahovať nasledujúce údaje:

- Celkové poradie pretekára
- Informáciu, či sa poradové miesto pretekára oproti predošlým kolám zlepšilo, zhoršilo alebo nezmenilo
- Meno a priezvisko
- Registračný kód
- Bodový zisk pre každé z doterajších kôl (v prípade, že sa ho pretekár zúčastnil)
- Celkový bodový zisk

1.3.6.2 Vygenerovanie súboru

*A

1.3.6.2.1 Súbor sa vygeneruje, keď používateľ v programe zvolí možnosť “vygenerovať priebežné hodnotenie sezóny”.

1.3.6.2.2 Súbor sa vygeneruje po každom pridaní vstupného súboru.

1.3.7 Validácia

Validácia výsledného softvérového diela bude prebiehať kontrolou praktickej využiteľnosti programu.

1.3.7.1 Nastaviteľná konfigurácia

Zložka projektu bude okrem samotného programu a iných potrebných častí obsahovať aj konfiguračný súbor. V tomto súbore sa budú nachádzať všetky premenlivé hodnoty, na základe ktorých bude fungovať algoritmus vypočítavajúci rozdelenie bodov:

1.3.7.1.1 Rok sezóny

*A

Do konfiguračného súboru používateľ zadá rok sezóny.

1.3.7.1.2 Počet bodovaných kôl

*A

Do konfiguračného súboru používateľ zadá maximálny počet kôl, za ktoré sa pretekárovi počítajú body do celkového rebríčka. Pretekár sa môže zúčastniť aj viacej kôl, v tom prípade sa mu najhoršie bodové zisky nezapočítavajú. Predvolená hodnota je 10 kôl.

1.3.7.1.3 Spôsob bodovania pretekov

*B

V konfiguračnom súbore bude ďalej možnosť uviesť, ktorý z vyššie uvedených (požiadavka 3.5.3 „Zmena bodovania kôl“) spôsobov bodovania bude použitý na výpočet bodov. Toto sa vyberie zadáním číslu 1, 2 alebo 3. V prípade použitia prvého spôsobu výpočtu (3.5.2), sa použije zoznam, uvedený v konfiguračnom súbore. Tento zoznam sa skladá z ľubovoľného počtu čísel oddelených znakom ‘,’. V prípade použitia tretieho spôsobu bodovania (3.5.3.2) sa použije konštanta uvedená na konci konfiguračného súboru. Jej predvolená hodnota je 50.

1.3.8 *Prílohy*

1.3.8.1 Výstupný súbor

1.3.8.1.1 výstupný súbor po 7.kole v sezóne 2018:

http://zapad.orienteeering.sk/doc/2018/OR2018_07.pdf

2 Návrh softvéru

2.1 Úvod

2.1.1 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.

2.2 Špecifikácia vonkajších interfejsov

2.2.1 Pridanie vstupného súboru

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

2.2.2 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.

2.3 Dátové modely

2.3.1 Model perzistentných údajov

Stromová štruktúra podpriečinkov priečinka „DATA_STORAGE“ (v zložke so zdrojovými súbormi) a súborov:

- DATA_STORAGE
 - 2018
 - kolo1.xml
 - kolo2.xml
 - ...
- mainTemplate.txt
- ...
- vystup1.html
- vystup2.html
- ...

2.3.2 Model formátov súborov

2.3.2.1 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.orienteering.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>
```

2.3.2.2 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.

2.3.2.3 Formát šablón

Všetky šablóny sú formátu .txt. Slúžia na vytvorenie výstupného súboru. Šablóny môžu obsahovať nasledujúce kľúčové slová ktoré sa nahrádzajú:

- **@bodovanie** bude nahradené reťazcom podľa typu bodovania napríklad 50-45-40-35-30-29-28-27-...3-2-1-1-1-1... DISK = 0.
- **@pocetVysledkov** bude nahradené reťazcom ktoré vyjadruje počet kôl ktoré z ktorých sa výsledné body započítavajú do konečného výsledku
- **@navigacia** bude nahradené odkazmi na kategórie podľa šablóny navTemplate.html
- **@poradieKol** bude nahradené poradovými číslami kôl tvaru <td>poradie</td>
- **@datumyKol** bude nahradené dátumami konania kôl tvaru <td>dátum</td>
- **@vysledky** bude nahradené tabuľkami, podľa počtu kategórii a šablóny categTemplate.html
- **@kategoria** bude nahradené názvom kategórie napríklad M-10
- **@nazvyKol** bude nahradené názvami kôl tvaru <td>názov</td>
- **@prazdneStlpceKol** bude nahradené <td></td> podľa počtu kôl
- **@vysledkyBezcov** bude nahradené podľa šablóny runnerTemplate.html
- **@bodyKol** bude nahradené bodmi kôl tvaru <td>body</td>
- **@poradie** bude nahradené poradím pretekára
- **@zmena** bude nahradené zmenou oproti predošlému kolu pretekára (ak sa posunul o X priečok v tabuľke vyššie: ↑X, ak sa posunul o X priečok nižšie: ↓X, ak ostal na rovnakej pozícii: ↔)
- **@meno** bude nahradené menom pretekára
- **@priezvisko** bude nahradené priezviskom pretekára
- **@id** bude nahradené id-čkom pretekára
- **@spolu** bude nahradené počtom bodov pretekára z celej súťaže

mainTemplate.txt:

```
<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> <b>Sanasport Cup 2018 <br>Západná oblasť SZOŠ</b> </section>
      
    </div>
    <div>
      <br>Bodovanie: @bodovanie.
      <br>Počíta sa @pocetVysledkov najlepších výsledkov
    </div>
    <div>
      <br>Informácie a reklamácie: <i>Richard Balogh SKS6901</i>
      &lt;<b>balogh@elf.stuba.sk</b>&gt;
    </div>
  </header>
  <nav>
    @navigacia
  </nav>
  <table>
    <tr>
      <td></td> <td></td> <td></td>
      <td></td> <td></td> @poradieKol
      <td></td>
    </tr>
    <tr>
      <td></td> <td></td> <td></td>
      <td></td> <td></td> @datumyKol
      <td></td>
    </tr>
    <tr>
      <td></td> <td>Zmena oproti predošlému kolu</td> <td></td>
      <td></td> <td></td> @nazvyKol
      <td>Spolu</td>
    </tr>
    @vysledky
  </table>
</body>
</html>
```

navTemplate.txt:

```
<tr id = "#@kategoria">
  <td></td>
  <td></td>
  <td>@kategoria </td>
  <td></td>
  <td></td>
  @prazdneStlpceKol
  <td></td>
</tr>
@vysledkyBezcov
```

categTemplate.txt:

```
<a href="#@kategoria">@kategoria</a>
```

runnerTemplate.txt:

```
<tr>
<td>@poradie</td>
<td>@zmena</td>
<td>@meno</td>
<td>@priezvisko</td>
<td>@id</td>
@bodyKol
<td>@spolu</td>
```

2.3.2.4 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>
```

2.4 Návrh používateľského rozhrania

2.4.1 Návrh interakcie programu cez konzolu s používateľom

Interakciu s používateľom na konzole zaobstaráva *Interakčný modul*.

2.4.1.1 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.

2.4.1.2 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 súbormi*, ktorý s ňou ďalej pracuje.

2.4.1.3 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.

2.5 Rozdelenie na moduly a ich interfejsy

2.5.1 Spúšťačí modul

Modul zabezpečuje spustenie programu a naštartovanie kľúčových modulov.

2.5.1.1 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.

2.5.2 Interakčný modul

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

2.5.2.1 Spracovanie dát

2.5.2.1.1 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.

2.5.2.1.2 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 modulu, ktorý neúplnosť odhalilo.

2.5.2.2 Výstupné dáta

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

2.5.2.3 Kontrola Vstupov

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

2.5.2.3.1 Kontrola formátu

Pri kontrole sa prihliada na formát údaju. Modul zabezpečuje odmietnutie vstupného údaju chybného formátu.

2.5.2.3.2 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).

2.5.2.4 Rozhranie Interakčného modulu

Interakčný modul poskytuje všetkým ostatným modulom rozhranie prostredníctvom statických metód:

2.5.2.4.1 Rozhranie s *modulom na prácu so vstupnými súbormi*

- **String getInputFileAddress()**
 - metóda, pomocou ktorej modul na prácu so vstupnými súbormi získa adresu vstupného súboru od interakčného modulu

2.5.2.4.2 Rozhranie s *modulom na kontrolu vstupného súboru*

- **Integer requestRaceRank()**
 - metóda na vyžiadanie dodatočného údaj (číslo kola) v prípade, že tento údaj nie je uvedený vo vstupnom súbore
- **String requestRaceName()**
 - metóda na vyžiadanie dodatočného údaj (názvu kola) v prípade, že tento údaj nie je uvedený vo vstupnom súbore
- **String requestRaceStartTime()**
 - metóda na vyžiadanie dodatočného údaj (času začiatku kola) v prípade, že tento údaj nie je uvedený vo vstupnom súbore

2.5.2.4.3 Rozhranie so všetkými ostatnými modulmi

- **printMessage(String message)**
 - metóda na výpis akejkoľvek správy (upozornenia) na konzolu

2.5.3 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.

2.5.3.1 Vstupné údaje

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

Pri načítavaní všetkých vstupných súborov modul nemá zadávané vstupné dáta.

2.5.3.2 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.

2.5.3.3 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_STORAGE, podpriečinka s názvom rovným stringovej reprezentácii hodnoty premennej seasonYear.

Následne načíta všetky vstupné súbory. (Program pokračuje procesom počítania bodov)

2.5.3.4 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_STORAGE 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.

2.5.3.5 Výstupné údaje

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.

2.5.3.6 Rozhrania

Modul poskytuje rozhrania:

addInputFile(String inputFileAddress) – pridá vstupný súbor a spustí načítanie všetkých vstupných súborov (metóda loadInputFiles())

loadInputFiles() – načíta do objektov všetky vstupné súbory sezóny a odošle ich *Výpočtovému modulu*

2.5.4 Modul na kontrolu vstupného súboru

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

2.5.4.1 Vstupné údaje

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

2.5.4.2 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.

2.5.4.3 Výstupné údaje

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

2.5.4.4 Rozhrania

Modul poskytuje jedno rozhranie pre *Modul na prácu so vstupnými súbormi*.

ResultList verifyInputFile(ResultList resultList) – vráti skontrolovaný objekt triedy ResultList zadany ako argument.

2.5.5 Konfiguračný modul

2.5.5.1 Vstupné dáta

Program dostane adresu konfiguračného súboru.

2.5.5.2 Spracovanie dát

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

2.5.5.3 Výstupné dáta

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

2.5.6 Výpočtový modul

2.5.6.1 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.

2.5.6.2 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 nastaví body objektom triedy PersonResult, nič nevracajú.

2.5.6.3 Výstupné dáta

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

2.5.7 Priradovací modul

2.5.7.1 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.

2.5.7.2 Spracovanie dát - trieda TotalResults

Trieda TotalResults používa objekty triedy ResultList, ktoré postupne prechádza a priradzuje potrebné údaje do slovníka. Tento slovník obsahuje skratky kategórií ako kľúče a ako hodnoty ďalšie slovníky, kde kľúčom je identifikátor pretekára a hodnota objekt triedy RunnerOverall(dalej popísanej). Identifikátor pretekára tvorí primárne id. Ak hráč nemá id, identifikátor má tvar “#priezvisko meno”.

Ak kategória ešte v slovníku nie je, vytvorí sa nový kľúč so skratkou kategórie. Ak v slovníku ešte nie je pretekár, do vnoreného slovníku sa pridá nový kľúč tvoriaci textový reťazec identifikujúci pretekára a ako hodnota príslušný objekt triedy RunnerOverall.

Táto trieda obsahuje aj zoznam dátumov jednotlivých kôl a zoznam reťazcov events, ktorý obsahuje chronologicky zoradené názvy jednotlivých kôl.

2.5.7.3 Spravovanie dát - trieda RunnerOverall

Trieda RunnerOverall obsahuje zoznam bodov pretekára za jednotlivé kola. Ak sa pretekár kola nezúčastnil, hodnota je 0. Ak sa zúčastnil, ale bol diskvalifikovaný, hodnota je -1. Atribút points obsahuje celkový súčet bodov pretekára, zohľadňujúci maximálny počet kôl, ktoré sa pretekárovi bodujú. Trieda udržiava aj atribúty pretekára - id, givenName a FamilyName.

2.5.7.4 5.7.4 Výstupné dáta

Modul odovzdá objekt triedy TotalResults *Generujúcemu modulu*.

2.5.8 Generujúci modul

2.5.8.1 Vstupné dáta

Modul dostáva objekt triedy TotalResults obsahujúci slovník celkových výsledkov od *Prirad'ovacieho modulu* a pracuje aj s predpripravenými šablónami.

2.5.8.2 Spracovanie dát

Modul načíta pripravené šablóny do premenných triedy HtmlResultGenerator mainTemplate, navTemplate, categoryTemplate a runnerTemplate. Trieda obsahuje funkcie fillMainTemplate, fillNavTemplate, fillCategoryTemplate a fillRunnerTemplate ktoré podopĺňajú dáta zo slovníka obsahujúceho objekty triedy TotalResults a vrátia podopĺňané stringy ktoré sa zapíšu do výsledného html súboru.

2.5.8.3 Výstupné dáta

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

2.6 Návrh implementácie

2.6.1 Use-case diagram



2.6.2 UML component diagram

Diagram je priložený na adrese diagrams/component_diagram.png

2.6.3 UML class diagram

Diagram je priložený na adrese diagrams/class_diagram.png

2.6.4 UML sequence diagram

Diagram je priložený na adrese diagrams/sequence-diagram.png

2.6.5 Použité technológie

2.6.5.1 Java, verzia 8

Programovací jazyk pre ktorý je program skompilovaný.

2.6.5.2 Java Architecture for XML Binding

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

V programe sú použité knižnice:

com.sun.xml.bind:jaxb-impl:2.3.1

com.sun.xml.bind:jaxb-core:2.3.0

javax.xml.bind:jaxb-api:2.3.1

3 Testovacie scenáre

3.1 Testovacie scenáre pre pridávanie vstupného súboru

package TestsInputFileHandling

rebricek -bezcov\src\TestsInputFileHandling\TestInputFileHandling.java

zoznam unit testov:

- a) *testAddNotXML* – požiadavka [1.12.2.2]
pridanie súboru, ktorý nie je formátu .xml (TestsInputFiles/notxmlfile.txt)

- b) *testLoadNonexDir* – načítanie uložených vstupných súborov z neexistujúceho priečinka DATA_STORAGE
- c) *testCreateCopyAndFolder* – požiadavky [1.11.1], [1.11.2.5]
pridanie správne zadaného vstupného súboru, vytvorenie priečinku DATA_STORAGE/ a v ňom priečinku 2019/
- d) *testLoadSingleInputFile* – požiadavka [1.12.2.1]
pridanie správne zadaného vstupného súboru do existujúceho priečinku 2019
a overenie správnosti počtu načítaných kategórií v kole a celkového počtu pretekárov
- e) *testCreateCopies* – požiadavky [1.11.2.4], [1.12.1]
pridanie 3 správne zadaných vstupných súborov (majú aj rozdielne Round)
- f) *testFillEvent* – požiadavky [1.11.2.3], [3.1.1.1]
pridanie neúplného vstupného súboru, chýbajú informácie v Event, doplnenie údajov
a overenie správnosti doplnených údajov.

3.2 Testovacie scenáre pre inicializáciu programu

- **Spustenie: Variant A**
 - Test na správne vyhodnotenie prepínača A pri spustení z konzoly
- **Spustenie: Variant B**
 - Test na správne vyhodnotenie prepínača B pri spustení z konzoly
- **Spustenie: Variant C**
 - Test na správne vyhodnotenie pri spustení z konzoly bez prepínača, alebo spustení cez ikonu. Test overuje, že program správne vyhodnotí typ spustenia a dá používateľovi na výber z variantov A a B.
- **Spustenie: Variant C, vyber A**
 - Test na správne vyhodnotenie výberu varianty A z výberu, poskytnutom po zvolení variantu C.
- **Spustenie: Variant C, vyber B**
 - Test na správne vyhodnotenie výberu varianty B z výberu, poskytnutom po zvolení variantu C.
- **Spustenie: Variant C, zly vyber**
 - Test na správne vyhodnotenie výberu nevalidnej varianty z výberu, poskytnutom po zvolení variantu C. Test overuje, že program opätovne vyžiada výber varianty A/B.
- **Spustenie: Zly variant**
 - Test na správne vyhodnotenie nevalidného prepínača pri spustení z konzoly. Test overí, že program sa zachová, ako pri výbere varianty C a dá používateľovi na výber z variant A a B.

3.3 Testovacie scenáre pre interakciu

- **Kontrola vstupov: Adresa vstupneho suboru Pozitivny**

- Test na správnu kontrolu adresy vstupného súboru. Zadaná adresa je správna, test overí, že validácia ju vyhodnotí, ako správnu.
- **Kontrola vstupov: Adresa vstupneho suboru Negativny – Format**
 - Test na správnu kontrolu adresy vstupného súboru. Zadaná adresa má nesprávny formát. Test overí, že validácia ju vyhodnotí ako nesprávnu, upozorní na túto skutočnosť a opätovne vyžiada od používateľa zadanie adresy vstupného súboru.
- **Kontrola vstupov: Adresa vstupneho suboru Negativny – Relevance**
 - Test na správnu kontrolu adresy vstupného súboru. Zadaná adresa má správny formát, ale vedie k súboru, ktorý neexistuje. Test overí, že validácia ju vyhodnotí ako nesprávnu, upozorní na túto skutočnosť a opätovne vyžiada od používateľa zadanie adresy vstupného súboru.
- **Poskytovane rozhrania: PrintMessage**
 - Test na správnu funkcionálnosť metódy na vypisovanie upozornení
- **Poskytovane rozhrania: RequestRaceRank Pozitivny**
 - Test na vyžiadanie dodatočného údaju o preteku – čísla kola preteku. Test overí správne vyžiadanie údaju a správne spracovanie. Zadané číslo kola má správny formát, program toto zadané číslo akceptuje.
- **Poskytovane rozhrania: RequestRaceRank Negativny**
 - Test na vyžiadanie dodatočného údaju o preteku – čísla kola preteku. Test overí správne vyžiadanie údaju a správne spracovanie. Zadané číslo kola má nesprávny formát, program toto zadané číslo neakceptuje, upozorní na skutočnosť o chybnom formáte a opätovne vyzve používateľa na zadanie údaju.
- **Poskytovane rozhrania: RequestRaceName**
 - Test na vyžiadanie dodatočného údaju o preteku – názvu kola preteku. Test overí správne vyžiadanie údaju, správne spracovanie a uloženie.
- **Poskytovane rozhrania: RequestRaceStartTime Pozitivny**
 - Test na vyžiadanie dodatočného údaju o preteku – času začiatku kola preteku. Test overí správne vyžiadanie údaju a správne spracovanie. Zadaný čas začiatku kola má správny formát, program tento čas akceptuje.
- **Poskytovane rozhrania: RequestRaceStartTime Negativny**
 - Test na vyžiadanie dodatočného údaju o preteku – času začiatku kola preteku. Test overí správne vyžiadanie údaju a správne spracovanie. Zadaný čas začiatku kola má nesprávny formát, program tento čas neakceptuje, upozorní na skutočnosť o chybnom formáte a opätovne vyzve používateľa na zadanie údaju.
- **Poskytovane rozhrania: getIpAddress**
 - Test na správne poskytnutie adresy vstupného súboru inému modulu. Test zahŕňa aj správne zadanie, spracovanie a uloženie adresy vstupného súboru.