



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta strojní



APLIKACE PRO PODPORU POŘÁDÁNÍ ZÁVODŮ PRO ORIENTAČNÍ SPORTY

Diplomová práce

Studijní program: N2301 – Strojní inženýrství
Studijní obor: 3902T021 – Automatizované systémy řízení ve strojírenství

Autor práce: **Bc. Tomáš Kobr**
Vedoucí práce: Ing. Michal Moučka, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechanical Engineering ■

APPLICATION FOR SUPPORT TO ORGANIZE COMPETITIONS FOR ORIENTEERING SPORTS

Diploma thesis

Study programme: N2301 – Mechanical Engineering
Study branch: 3902T021 – Automated Control Systems

Author: **Bc. Tomáš Kobr**
Supervisor: Ing. Michal Moučka, Ph.D.



ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Tomáš Kobr**
Osobní číslo: **S12000446**
Studijní program: **N2301 Strojní inženýrství**
Studijní obor: **Automatizované systémy řízení ve strojírenství**
Název tématu: **Aplikace pro podporu pořádání závodů pro Orientační sporty**
Zadávací katedra: **Katedra výrobních systémů a automatizace**

Z á s a d y p r o v y p r a c o v á n í :

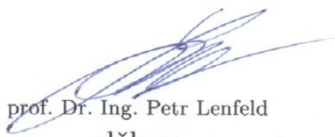
1. Navrhněte strukturu aplikace pro pořádání závodů v orientačním běhu. Pro práci s daty využijte některý z volně dostupných databázových systémů. Aplikace musí pořadatelům umožnit:
 - a) komunikaci (import a export dat) s centrálním systémem ORIS českého svazu orientačních sportů (ČSOS),
 - b) komunikaci s elektronickou časomírou SPORTident,
 - c) předzávodní editaci dat, losování časů pro intervalové starty, zpracování výsledků.
2. Aplikaci naprogramujte.
3. Ověřte funkčnost a správnost řešení.

Rozsah grafických prací: **dle potřeby**
Rozsah pracovní zprávy: **cca 45 stran + přílohy**
Forma zpracování diplomové práce: **tištěná/elektronická**
Seznam odborné literatury:

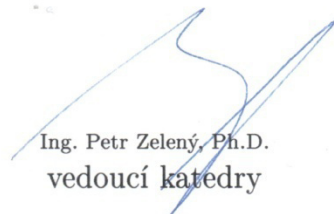
- [1] *C# Reference*. <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx> (accessed Oct 29, 2014).
- [2] *C# Programming Guide*. <http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx> (accessed Oct 29, 2014).
- [3] HUDDLESTON, J. a V.V. AGARWAL. *Databáze v C# 2008*. Praha: Computer Press, 2009. ISBN 978-80-251-2309-6.
- [4] NAGEL, CH., B. EVJEN, J. GLYNN a M. SKINNER a K. WATSON. *C# Programujeme profesionálně 2008*. Praha: Computer Press, 2013. ISBN 978-80-251-2401-7.
- [5] SPORTIdent. *PC-Programmes Guide*, 2013.

Vedoucí diplomové práce: **Ing. Michal Moučka, Ph.D.**
Katedra výrobních systémů a automatizace

Datum zadání diplomové práce: **5. března 2015**
Termín odevzdání diplomové práce: **5. června 2016**


prof. Dr. Ing. Petr Lenfeld
děkan




Ing. Petr Zelený, Ph.D.
vedoucí katedry

V Liberci dne 5. března 2015

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Aplikace pro podporu pořádání závodů pro Orientační sporty

Application for support to organize competitions for Orienteering sports

Anotace

Diplomová práce se zabývá tvorbou aplikace pro pořádání závodů v orientačních sportech. V první části jsou vysvětleny základní pojmy týkající se orientačních sportů se zaměřením na orientační běh a informační systém ORIS. Kratší část je věnována popisu některých podobných existujících programů. Hlavní část popisuje kompletní návrh celé aplikace. V závěru je demonstrováno testování vytvořeného softwaru.

Klíčová slova: orientační sporty, informační systém pro orientační sporty, databáze, programovací jazyk C#, systém SPORTident

Annotation

This diploma thesis deals with creating of the application for organizing orienteering competitions. The first part of the thesis describes basic concepts of orienteering sports, especially foot orienteering and czech information system ORIS. The next part represents some similar existing applications. The main part is about the whole concept of the newly created application. The last part of the thesis shows functionality of the application during testing on a public race.

Keywords: orienteering sports, information system for orienteering sport, database, C#, SPORTident system

Poděkování

Tímto bych rád poděkoval Ing. Michalovi Moučkovi, Ph.D. za odborné vedení diplomové práce a poskytnutí cenných rad při její tvorbě. Chtěl bych poděkovat Ing. Martinovi Váňovi za náměty a připomínky týkajících se aplikací pro orientační sporty a zkušené rady při komunikaci se systémem SPORTident. Taktéž bych rád poděkoval pořadatelům závodu OCAD cup, kteří mi během něj umožnili otestovat vytvořenou aplikaci. Oddílům Slavia Liberec orienteering, OOB TJ Tatran Jablonec a OOB TJ Turnov děkuji za vypůjčení materiálu k testování aplikace. V neposlední řadě děkuji svým rodičům za podporu, bez které by tato práce nemohla vzniknout.

OBSAH

	SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	9
1	ÚVOD	10
2	ORIENTAČNÍ SPORTY	12
	2.1 Orientační běh a vybavení závodníka	12
	2.2 Vývoj kontrolních jednotek	13
	2.3 Popis a pravidla závodní trati	13
	2.4 Scorelauf	14
	2.5 Způsoby startování	15
	2.6 Vakant	16
3	EXISTUJÍCÍ SYSTÉMY	17
	3.1 Systém OB2000	17
	3.2 Systém OE2010	17
4	INFORMAČNÍ SYSTÉM PRO ORIENTAČNÍ SPORTY	19
5	NÁVRH DATABÁZE	21
	5.1 Databáze – základní pojmy	21
	5.2 Výběr databáze	22
	5.3 Připojení databáze	22
	5.4 Ošetření otevřené databáze	23
	5.5 Jazyk SQL	23
	5.6 Manipulace s daty	25
	5.7 Struktura navržené databáze	26
6	SPUŠTENÍ APLIKACE	28
7	ZALOŽENÍ NOVÉHO ZÁVODU	30
	7.1 Základní informace	30
	7.2 Načtení závodníků	30
	7.2.1 Načtení dat ze systému ORIS	31
	7.2.2 Načtení dat z textového souboru	33
	7.2.3 Ruční načtení	34
	7.3 Dokončení a nahrání do databáze	34
8	EDITACE ZÁZNAMŮ	36
	8.1 Změna parametru – buňky	36
	8.2 Editace záznamu VAKANT	38
	8.3 Přidání záznamu	40
	8.4 Odebrání záznamu	40
	8.5 Aktualizace záznamů	40
	8.6 Vyhledávání parametru	41
	8.7 Změna závodníka na vakanta	41

	8.8 Spojení kategorií	41
	8.9 Rozdělení kategorií	42
	8.10 Rozdělení do kvalifikačních skupin	45
9	TRATĚ	47
	9.1 Vstupní kontrola	47
	9.2 Výstupní kontrola	47
	9.3 Ruční načtení tratí	48
	9.4 Načtení tratí ze souboru	48
	9.5 Úprava tratí	49
	9.6 Smazání trati	50
10	TVORBA STARTOVNÍCH ČASŮ	51
	10.1 Losování závodníků	54
11	SBĚR DAT Z ČIPŮ	56
	11.1 Systém SPORTident	56
	11.2 Závodní průkazy - čipy	56
	11.3 Komunikace se systémem SPORTident	57
	11.4 Zjištění konkrétní série čipu	58
	11.5 Načtení dat z čipu	59
	11.6 Úprava načteného záznamu	62
	11.7 Tisk mezičasů	63
	11.8 Nastavení tisku	65
12	VÝSLEDKY	67
13	EXPORTY DAT	68
	13.1 Export do PDF	68
	13.2 Export do HTML	69
	13.3 Export výsledků pro ORIS	69
14	TESTOVÁNÍ APLIKACE	70
15	ZÁVĚR	72
	SEZNAM POUŽITÉ LITERATURY	74
	SEZNAM PŘÍLOH	76

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

ČSOS	Český svaz orientačního běhu
dll	dynamicky linkovaná knihovna
DOS	diskový operační systém
Emit	norský systém elektronické časomíry
GPS	globální polohovací systém
html	značkovací jazyk, používaný pro tvorbu webových stránek
ID	identifikace
json	javascriptový objektový zápis
LOB	lyžařský orientační běh
MTBO	orientační závody na horských kolech
OB	orientační běh
Ocad	software pro tvorbu map a tratí
ORIS	informační systém českého orientačního běhu
pdf	přenosný formát dokumentů
RS-232	konektor sériového portu
sdf	databázový soubor
SI	zkrácený název pro systém SPORTident
SPORTident	německý systém elektronické časomíry
TRAIL-O	sportovní disciplína zaměřená na orientaci a na správné čtení a interpretaci mapy
txt	textový soubor
USB	univerzální sériová sběrnice
xml	rozšířitelný značkový jazyk

1 ÚVOD

Každý člověk, který se někdy podílel na pořádání určité společenské události, moc dobře ví, že ke zdárnému hodnocení je potřeba vynaložit potřebné úsilí. Velikost takového úsilí je ovlivněna různými faktory, ať už jde např. o typ události nebo počet zúčastněných osob. Pro organizování závodů v orientačních sportech je pro usnadnění práce možno využít počítačových programů.

Z pořadatelského hlediska můžeme závody v orientačních sportech rozdělit na dvě části. Za první jde o práci v terénu, kdy se řeší tvorba mapy, kontrolní stanoviště, zázemí závodů, různá povolení, catering atd. Druhou částí je pak zpracování dat různě souvisejících se samotným závodem - od osobních údajů přihlášených účastníků až po kontrolu správnosti výsledků pro všechny kategorie. Právě touto druhou částí se zabývá tato diplomová práce.

V současné době existují programy, které se využívají při pořádání závodů. Některé z nich jsou v diplomové práci představeny. V posledních letech jsem se začal o tento typ softwaru zajímat. V roce 2013 byl spuštěn informační systém pro orientační sporty – ORIS, což je komplexní webová databáze dat týkajících se závodů v orientačních sportech. Jelikož ORIS umožňuje exportovat různá data týkajících se závodů (např. přihlášek), naskytla se myšlenka vytvořit program, který by s touto databází komunikoval, což starší programy neumožňovaly. Díky této komunikaci by bylo výrazně zrychleno načtení dat do databáze programu. Pro komunikování s aplikací ORIS je potřeba mít internetové připojení, což v dnešní době není velká překážka.

Cílem diplomové práce je vytvořit funkční aplikaci, která plní potřeby pořádání závodů v orientačních sportech. Aplikace má mít pevně navrženou strukturu a vhodně zvolený databázový systém. Program musí umět komunikovat s webovou databází ORIS resp. umět sbírat a načítat data. Taktéž má být zajištěna komunikace se sportovní časomírou SPORTident. Aplikace má umožnit uživateli předstartovní editaci dat, losování startovních časů pro intervalové starty (dle pravidel ČSOS) a zpracování výsledků. Vytvořený program by měl být nakonec také otestován.

V úvodu je vysvětleno, jaká je podstata orientačních sportů. Krátce je vysvětlen vývoj kontrolních stanovišť. Představen je taktéž webový systém ORIS. Následuje hlavní část, počínaje vytvořením databázového modelu. Detailně jsou popsána jednotlivá okna programu, přesněji řečeno, jak fungují. K některým částem je ukázán a popsán zdrojový

kód programu, pro zachování přehlednosti práce jsem vybral kratší a důležité úseky kódu. Kompletní program je k dispozici na přiloženém médiu. Část se věnuje elektronické časomíře SPORTident – jejím specifikám, komunikaci s ní a práci se získanými daty. Závěr patří testování na veřejném závodě. Aplikace je naprogramována v jazyce C#.

2 ORIENTAČNÍ SPORTY

Orientační sporty jsou synonymem pro pohyb v přírodě s mapou, ať už jde o variantu na horském kole, běžeckých lyžích či po vlastních nohou. Na startu závodu obdrží závodník vytištěnou mapu se zákresem tratě, kterou se snaží zdolat v co nejkratším možném čase. Trať obsahuje množství kontrolních stanovišť – kontrol. Aby mohl být účastník klasifikován, musí projít všemi kontrolami v předem daném pořadí. Každá kontrola má svůj ojedinělý kód. K ověření správného kódu slouží sportovci posloupnost kontrol - popisy, které jsou natištěny na mapě nebo jsou volně k dispozici před závodem. Obsahují kromě kódu také detaily kontrolního stanoviště např. typ objektu, jeho velikost atp. Důležité je zdůraznit, že cestu mezi kontrolami si volí závodník sám (dle pravidel v daném odvětví) s výjimkou povinných úseků.

Speciální variantou orientačních sportů je tzv. TRAIL-O vhodné pro handicapované sportovce. Všechny zmíněné varianty mají svá oficiální mistrovství světa. Dá se říci, že se jedná o sport všech věkových kategorií.

2.1 Orientační běh a vybavení závodníka

Když se zaměříme na orientační běh, tak nejmladší jedinci začínají běhat s doprovodem. Jejich „lektoři“ jim během trati vysvětlují, co mapové značky zakreslené v mapě znamenají ve skutečnosti, jak probíhá označení kontroly atp. Samostatně závodit začínají zhruba od 10let.

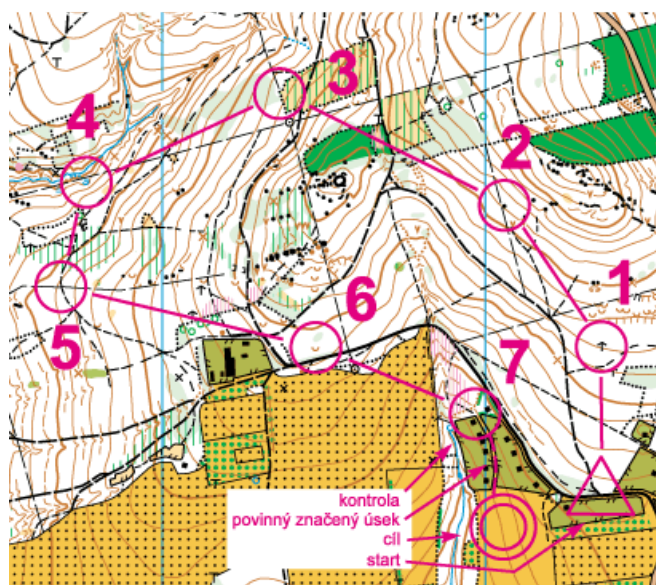
Výbava pro závody v orientačním běhu není většinou striktně daná, závodníci volí různý oděv, obuv a sportovní doplňky s ohledem na typ prostředí závodu. Pro lesní disciplíny je vhodné zakrytí dolních končetin a použití bot s hřebíky, naopak v městských disciplínách se volí volnější oblečení a boty bez hřebíků. Z hlediska pomůcek musí mít účastník závodní průkaz pro označení průchodu kontrolami. K navigaci používá mapu obdrženou na startu, popisy kontrol a buzolu. Závod by taktéž měl absolvovat se startovním číslem umístěným na hrudi, pokud jej od pořadatele obdrží. Pokud se jedná o noční závod, je zapotřebí svítilna. Další navigační přístroje jako telefon, GPS jsou zakázány s výjimkou pasivního GPS přístroje pro zaznamenávání trati.

2.2 Vývoj kontrolních jednotek

První závod se uskutečnil v roce 1897 norském Oslu. Tehdy se zúčastnilo 8 závodníků na trati dlouhé 10,5 km. Prvním systémem zaznamenávání průchod kontrolou, byl otisk razítka do přiděleného průkazu. Později přišel systém mechanických kleští. Jedná se o princip, kdy kleště mají různý počet vpichů. Závodník kontroly označí do daných míst závodního průkazu. Pokud však orazí kontrolu do špatné kolonky, jsou k dispozici náhradní místa na kartě. V současnosti se již používají elektronické systémy. Převážně se jedná o systém SPORTident. Lze se také setkat se systémem Emit, používaný více ve skandinávských zemích. Software mnou vytvořený bude komunikovat se systémem SPORTident, poněvadž je určený pro české závody, které tento systém používají. Na kontrole je elektronická stanice ze sady používaného systému, ve které běží čas, jenž je synchronizován se všemi ostatními v závodě. Dále je v ní uchován kód kontroly. Úspěšný průchod kontrolou je signalizován jak zvukovým, tak i světelným signálem.

2.3 Popis a pravidla závodní trati

Startovní čára je označována jako tzv. hrubý start, od kterého se počítá čas, a od kterého je závodník povinen doběhnout k tzv. ostrému startu. Ten je v terénu vyznačen lampionem a v mapě značkou trojúhelníku.



Obr. 2.3-1: Mapa se závodní tratí pro orientační běh [6]

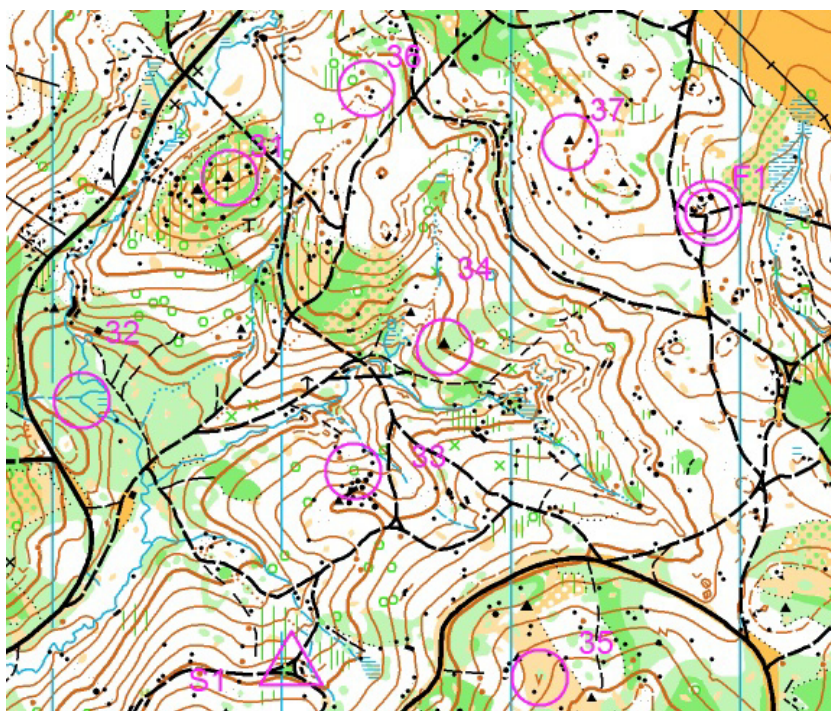
Cesta k ostrému startu bývá značena fáborcky. Jednotlivé kontroly na trati v mapě mají tvar kroužku. Ve skutečnosti je přítomen opět lampion se stojanem a v současné době s již připnutou elektronickou krabičkou, do které závodník vloží svůj čip. Pokud dojde k poruše elektronického systému, jsou na stanovišti k dispozici mechanické kleště, pomocí kterých označí kontrolu do mapy.

Kontroly jsou v mapě spojeny plnou čarou s výjimkou přerušované čáry, znamenající povinný úsek. Ten se typicky používá pro doběh z poslední kontroly do cíle. Poslední kontrole před cílem se říká sběrná kontrola a disponují jí všechny tratě v závodě. Cíl má v mapě tvar dvojitého kroužku. Poloha kontrolních stanic je v mapě umístěná doprostřed daného symbolu závodní trati.

Délka trati je součet všech úseků mezi kontrolami. Vzdálenost mezi kontrolami se počítá vzdušnou čarou. Kromě délky trati se uvádí její převýšení. Je to minimální velikost vystoupaných metrů ve vertikálním směru. Velikost převýšení je pouze informativní.

2.4 Scorelauf

Jedná se o speciální formu trati. Od klasické se liší tím, že skupinu kontrol lze absolvovat ve volném pořadí. Každá skupina musí mít vstupní a výstupní kontrolu.



Obr. 2.4-1: Scorelauf se vstupní (startovní) a výstupní (cílovou) kontrolou

2.5 Způsoby startování

V orientačním běhu je čas startu jednotlivých závodníků losován a před závodem zveřejněn na internetu. Na startu závodu jsou hodiny s časem pro informaci, kdy má závodník vstoupit do předstartovní zóny.

Intervalový start:

Druh startování, kdy závodník startuje osamoceně podle přiděleného času. Rozestup mezi dvěma závodníky je tzv. interval. Dle přiděleného času vstupuje závodník do předstartovní zóny. Tato zóna je složena zpravidla ze tří úseků. V prvním dochází ke kontrole čipu závodníka pořadatelem a též k vymazání starých dat v čipu. V dalším probíhá kontrola vynulování čipu a v posledním už jde o připravení se závodníka ke startu. V každém koridoru musí závodník setrvat minutu. Začátek každé minuty je dán zvukovým signálem. Vstupem do startovních koridorů, je čas hodin posunut vpřed právě o tolik minut, kolik je startovních koridorů, aby závodník vystartoval v přesně přiděleném čase.

Startovní krabíčka:

V překladu jde o volný start. Používá se především pro tréninkové (náborové) kategorie. Vstupem do předstartovní zóny v libovolném čase, je závodník pořadatelem vyslán speciálním koridorem až ke startovní čáře. Cestou je potřeba opět vymazat stará data z čipu a zkontrolovat vymazání. Na startovní čáře je připravena startovní krabíčka, která při označení zaznamená start závodníka.

Hromadný start:

Jde o typ startu, ve kterém startuje celá kategorie naráz. Závodníci se v předstartovní zóně seřadí např. podle startovních čísel.

Handicapovaný start:

Speciální typem startovní je tzv. stíhací závod – handicap. Je využitelný převážně při více-etapových závodech, kdy závodníci startují podle časových ztrát z předešlých etap.

2.6 Vakant

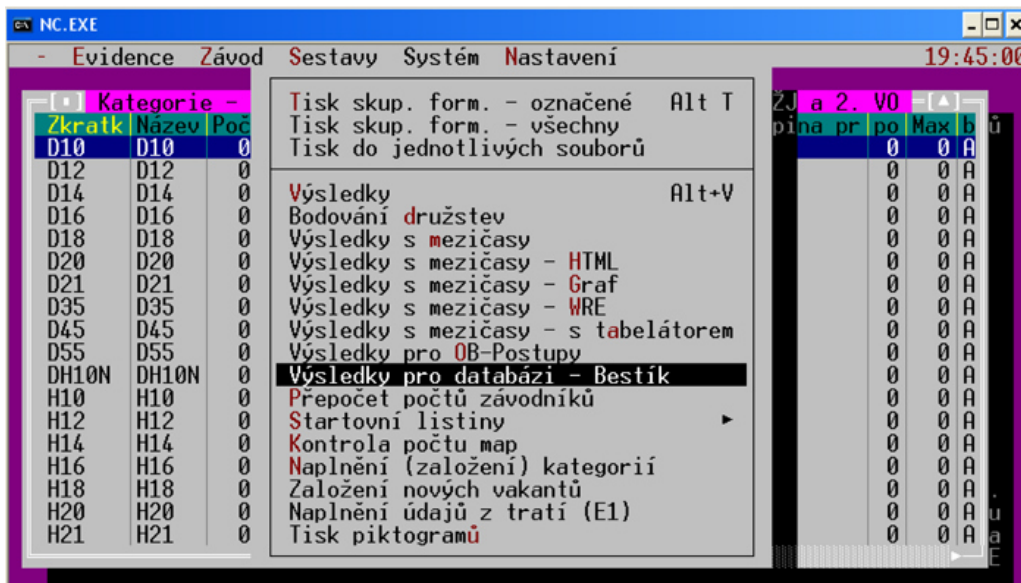
Speciální skupinou závodníků jsou tzv. vakanti. Jsou to virtuální závodníci, kteří reprezentují volná místa ve startovní listině příslušné kategorie. Počet vakantů garantuje pořadatel závodu. Tato volná místa jsou určena pro závodníky, hlásící se do závodu po termínu uzavření řádných přihlášek. Dohlášený závodník pak zaujme pozici příslušného vakanta společně s jeho startovním časem.

3 EXISTUJÍCÍ PROGRAMY

Spuštěním webové aplikace ORIS se nabízela otázka vytvoření programu pro pořádání závodů, který by navíc s ORISEM uměl komunikovat. Starší programy, které jsou využívány v České republice, komunikaci dosud nepodporovaly.

3.1 Systém OB2000

Aplikace vytvořená v roce 2001 Miroslavem Chmelařem, pro potřeby pořádání závodů v orientačním běhu s použitím SPORTidentu. Prvně byla vytvořena za účelem pořádání závodů pro Hanáckou oblast, později však byla využita nejen v České republice. Program prošel vývojem během několika let a spolehlivě funguje. Program lze využít nejen pro jednodenní závody, ale i pro vícedenní a štafetové závody. Nevýhodou této aplikace je skutečnost, že běží pod systémem DOS, což se projevuje jako estetická nevýhoda. Program taktéž neumí komunikovat s webovým systémem ORIS. Výhodou je, že jedná se volně dostupnou aplikaci.

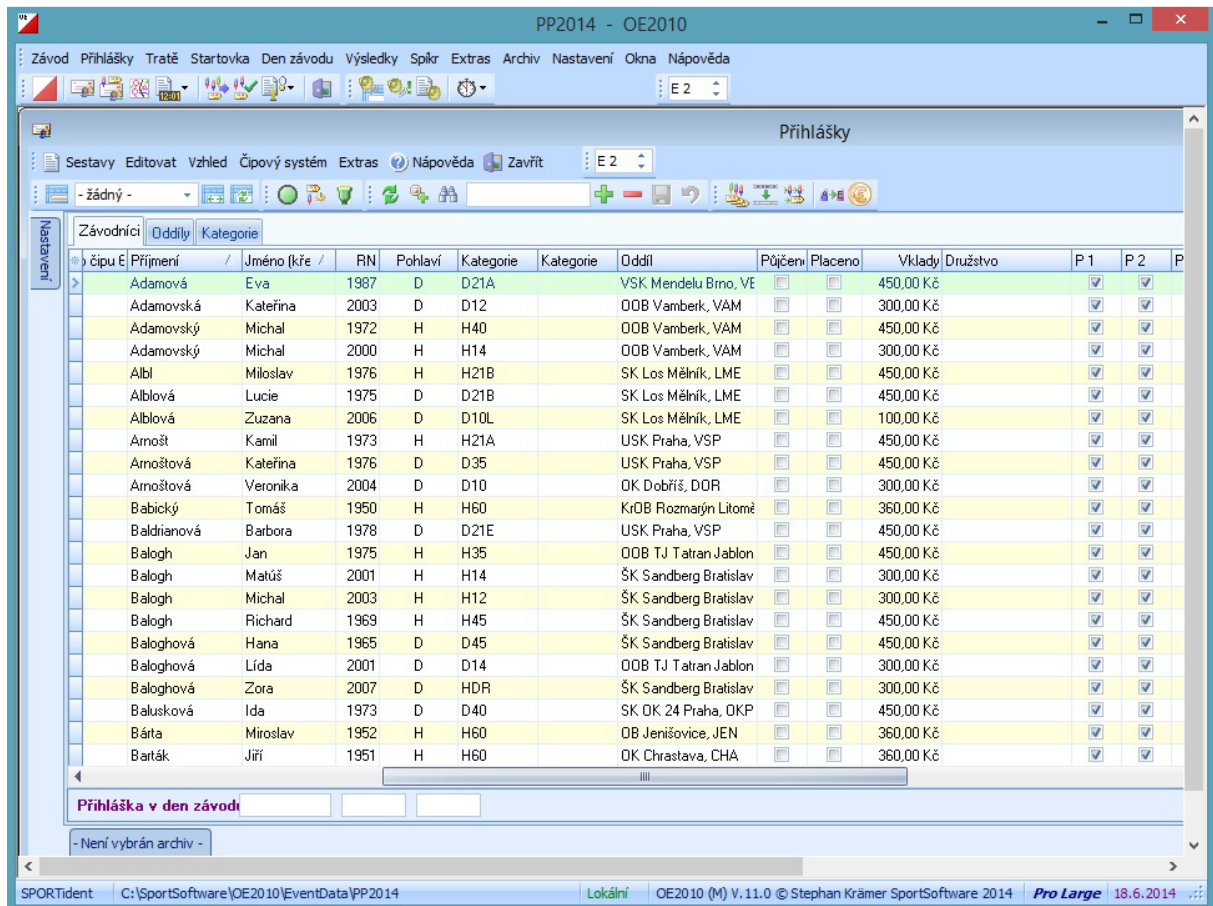


Obr. 3.1-1: Náhled do prostředí programu OB2000 [7]

3.2 Systém OE2010

Profesionální systém pro pořádání závodů, který vyvinul německý programátor Stephan Krämer. Počátek vývoje se váže k 70. létům minulého století. Jedná se zdroj obživy zmíněné osoby, pod jehož vedením se aplikace neustále vyvíjí. Z toho také vyplývá, že software není zdarma a jeho cena se liší podle daného typu (jednodenní a

vícedenní závody, štafety atp.). Cenové rozpětí se pohybuje mezi 15000 až 30000 korun. Jedná se o komplexní program, resp. lze využít různá nastavení (specifika závodění v odlišných státech), a proto je využíváný po celém světě. Z tohoto hlediska může aplikace působit nepřehledně. Do nynějška program neuměl komunikovat se systémem ORIS, nicméně neustálým vývojem je tato aktualizace údajně vytvořena.



Obr. 3.2-1: Náhled do prostředí programu OE2010

4 INFORMAČNÍ SYSTÉM PRO ORIENTAČNÍ SPORTY

Informační systém pro orientační sporty (ORIS) je centrální webový systém Českého svazu orientačních sportů (ČSOS), který pokrývá agendu vnitřních svazových procesů a agendu související se závody. Základními agendami systému jsou klubová evidence, registrace, přestupy, hostování, kalendář závodů, přihlášky, výsledky a další související informace. Hlavním autorem systému je Michal Besta, český orientační běžec a programátor, dlouhodobě pobývajícím ve Švédsku. První verze ORISu vznikla v roce 2010, přičemž vývoj probíhá až dodnes.

The screenshot shows the ORIS web application interface. At the top left is the ČSOS logo. On the right are navigation buttons: Kalendář, Adresář, Registrace, and Nápověda. Below the logo is a navigation menu with categories like Osobní, Nastavení, Mě závody, Mě kluby, Registrace, Hostování, Statistika, Licence k datu, Žebříčky, Ranking, Rozhodčí, Trenéři, Všechny kluby v systému, Exporty, Časté dotazy, and Helpdesk. The main content area displays a table of events for the year 2015. The table has columns: DATUM, NÁZEV, POŘ., REGION, SPORT, DIS., TYP, ZÁV., KLB., and P/V. Below the table are search filters for Datum, Název, Poř., Region, Sport, Dis., Typ, Záv., Klb., and P/V. The text 'Zobrazují 1 až 9 z celkem 9 záznamů' is visible below the filters.

DATUM	NÁZEV	POŘ.	REGION	SPORT	DIS.	TYP	ZÁV.	KLB.	P/V
21.02.2015	Mistrovství ČR na krátké trati	VPM	ČR	LOB	KT	MČR	248	3	
21.02.2015	Český pohár, ŽA, ŽB - sprint	VPM	ČR	LOB	SP	ČP	194	2	
22.02.2015	Český pohár, ŽA, ŽB	VPM	ČR	LOB	KL	ČP	234	2	
14.03.2015	Veřejný závod (Jarní skály)	TUV	JE	OB	KL	OF	162	0	07.03.
20.03.2015	Veřejný závod (Jarní pohár - noční)	PHK	VČ	OB	NOB	OF	10	0	08.03.
21.03.2015	Veřejný závod (Jarní pohár)	PHK	VČ	OB	KL	OF	28	0	08.03.
21.03.2015	Mistrovství oblasti v nočním OB	PGP	P, StČ	OB	NOB	OM	2	0	15.03.
21.03.2015	Mistrovství oblasti v nočním OB	AOP	MSK	OB	NOB	OM	2	0	17.03.
22.03.2015	Oblastní žebříček	PGP	P, StČ	OB	KT	OŽ	46	0	15.03.

Obr. 4-1: Náhled webového layoutu ORISU[8]

Pro náš účel je nejdůležitější sběr dat jednotlivých závodníků, přihlášených do příslušného závodu a údaje o kategoriích. ORIS dokáže, podle daných kritérií, vygenerovat údaje v souborech typu xml, json nebo txt. Tato aplikace si nechá vygenerovat žádané údaje v textových souborech a uspořádá si je vlastního souboru. Pro import výsledků je opět možnost využít několik typů struktur.

Příklad pro stažení konkrétních dat:

Přihlášky - <http://oris.orientacnisporty.cz/ExportPrihlasek?mode=ob2000&id=ORIS ID závodu>

Vložení tohoto odkazu do vyhledávače vyexportuje databáze soubor přihlášek závodníků. Za klíčovým slovem „id“ je zapotřebí použít číslo příslušného závodu. Získání ID závodu bude vysvětleno v kapitole 7.2.1.

5 NÁVRH DATABÁZE

5.1 Databáze – základní pojmy

Pod pojmem databáze si lze představit jakousi množinu dat, sloužící pro jejich uchování a manipulaci s nimi. Data se uchovávají uspořádaným a strukturovaným způsobem, usnadňující uživatelům jejich správu a získávání.

Entita - je objekt reálného světa (např. člověk, stroj, město), popsáný svými atributy a je jednoznačně odlišitelný od ostatních objektů.

Atribut - je prvek reálného světa, popisující entitu. Příkladem atributu pro entitu člověk může být jméno, příjmení, věk atp.

Vztahy - jsou spojitostí mezi entitami. Každá entita může mít vztah s více entitami najednou.

Relace - je tabulka, skládající se ze sloupců a řádků. Sloupce odpovídají příslušným vlastnostem (atributům) entity. Řádky jsou obvykle chápány jako záznamy. Soubor relací pak tvoří tzv. relační databázi.

Primární klíč - tabulky nemají nijak pevně dáno pořadí záznamů. K tomu, aby se dalo jednoznačně ukázat na jeden záznam, je nutné zvolit jeden atribut tabulky jako unikátní. Takovýto atribut je označován jako primární klíč.

Cizí klíč - Jedná se o speciální atribut, který se v tabulce vyskytuje, jsou-li záznamy v tabulkách mezi sebou propojeny vztahy. Do tohoto atributu se pak bude u jednotlivých záznamů podřízené tabulky kopírovat hodnota cizího klíče ze záznamu z nadřazené tabulky.

Příklad tabulky „závodník“, popisující entitu účastníka v závodě:

ID	JMÉNO	PŘÍJMENÍ	ODDÍL
1	Petr	Vejvoda	Tatran Jablonec
2	Luděk	Randýsek	OK Chrastava
3	Jakub	Volný	TJ Turnov

5.2 Výběr databáze

Pro náš účel jsem se rozhodl využít edice SQL serveru – SQL Server 2005 Compact Edition. Jedná se o balíček velikosti 3MB, který obsahuje knihovnu pro jednoduchou práci s databází. Dalo by se říci, že je to nástupce MS Accessu. Samostatná data se ukládají do souboru (s příponou SDF) a aplikace pracuje pomocí knihovny přímo s tímto souborem. Hlavní rozdíl mezi compact edicí a edicemi vyššími (počínaje Express edice) je ovšem ten, že compact edice neběží jako služba. Tímto lze říci, že uživatel nemusí instalovat SQL-server a také není zapotřebí nic konfigurovat. Je to víceméně jen knihovna, která otevře soubor a umožní pracovat s daty uvnitř souboru.

Databázový soubor lze využít jen z jedné aplikace v dané chvíli. To je ovšem její nevýhoda. V praxi však pro využití tohoto typu softwaru bývá pověřena jedna osoba = jedna aplikace. Při větších událostech může být využito i více osob, avšak si myslím, že vhodným ošetřením programu lze tento problém vyřešit, protože samostatné operace s daty trvají zanedbatelný čas. Edice compact slouží jako úložiště dat pro řádově tisíce záznamů. Komunikace s databázovým souborem probíhá pomocí SQL příkazů.

5.3 Připojení databáze

K práci s databází compact je zapotřebí vložit její knihovnu. K jejímu připojení je pak potřeba vypsát tento blok kódu:

```
using System.Data.SqlServerCe; // knihovna
using (SqlCeConnection cn = new SqlCeConnection("Data Source=c:\database.sdf"))
{
    cn.Open();
    /*
     * práce s databází
     */
    cn.Close();
}
```

Do konstruktoru třídy *SqlCeConnection* vypíšeme cestu k našemu souboru. Metoda *cn.Open* otevře databázi, následně už pracujeme se samotnými daty. K zavření databázového souboru slouží metoda *cn.Close*. Soubor je možno mít kdekoliv na disku. Pomocí třídy *OpenFileDialog* uživatel vybere žádaný soubor a následně je načtena adresa zvoleného souboru.

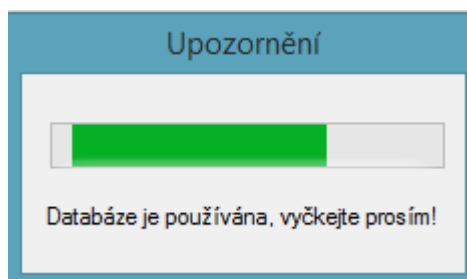
5.4 Ošetření otevřené databáze

Jak už bylo napsáno, otevřenou databázi je možno otevřít v danou chvíli jen z jedné aplikace. Proto bylo potřeba vytvořit ošetření k zachycení tohoto problému.

```
public void openDatabase(SqlCeConnection cn)
{
    Try
    {
        cn.Open();
    }

    catch
    {
        ErrorDialog pr = new ErrorDialog(connectionstring, cn);
        pr.ShowDialog();
    }
}
```

Metoda *openDatabase* se pokusí otevřít databázi. Pokud se jí nepodaří otevřít, objeví se dialogové okno, kde je uživateli vypsán text, že k databázi se nyní nelze připojit. V dialogu běží smyčka, běžící do doby, dokud se nepodaří k databázi připojit. Jakmile se podaří připojit, smyčka se ukončí a tím se zavře i dialog. Pokud připojení k databázi trvá déle než 10 sekund, je nastaven časový limit, který požadované spojení přeruší.



Obr. 5.4-1: Dialogové okno se zprávou o využívaném databázovém souboru

5.5 Jazyk SQL

Pro vykonávání databázových dotazů je využito syntaxe jazyka SQL. Ten obsahuje celou škálu příkazů, které by vydaly na celou diplomovou práci. Pro informaci uvedu jen několik z nich, potřebné k tomuto programu.

Získávání dat

Svou nejjednodušší podobu tvoří dotaz ze dvou částí:

```
SELECT <sloupec>
```

```
FROM <tabulka>
```

Chceme-li vybrat všechny sloupce z dané tabulky, použijeme symbol „*“. Příklad dotazu pro výběr všech atributů z tabulky závodník:

```
SELECT * FROM zavodnik
```

K výběru určitého sloupce je potřeba vypsát název atributu. Pro potřebu více sloupců, oddělíme názvy čárkou. Příklad výběru atributu jméno a příjmení z tabulky závodník:

```
SELECT jmeno,prijmeni FROM zavodnik
```

Vkládání dat

Nejjednodušší příkaz je složen z částí:

```
INSERT INTO <tabulka> (<sloupec>)
```

```
VALUES (<hodnota>)
```

Za názvem tabulky je potřeba do závorky napsat seznam názvů atributů, do kterých chceme vložit data, oddělené čárkou. Obsahuje-li tabulka primární klíč, musí být takto označená položka vložena. Lze se setkat i s inkrementovaným primárním klíčem, který je vložen automaticky. Příklad vložení jména a příjmení do tabulky závodník:

```
INSERT INTO zavodnik(jmeno, prijmeni) VALUES ('Tomáš', 'Kobr')
```

Klauzule WHERE

V dotazech je možno využít klauzule *WHERE*, umožňující stanovit určitá kritéria. Jednoduchým příkladem může být např. výběr všech řádků tabulky závodník, kde je v atributu pohlaví závodníka uveden muž:

```
SELECT * FROM zavodnik WHERE pohlavi='muž'
```

Aktualizace dat

Další důležitou činností je změna dat, složená z částí:

```

UPDATE    <tabulka>
SET       <sloupec> = '<hodnota>'
WHERE    <sloupec> = '<hodnota>'

```

Při vypisování dotazu je potřeba nezapomenout na klauzuli WHERE. Pokud by chyběla, aktualizovaly by se všechny řádky tabulky. Příklad změny čísla identifikačního průkazu v tabulce závodník, kde se účastníkovi rovná příslušné id:

```
UPDATE zavodnik SET cip='1969696' WHERE id='TJN8901'
```

Mazání dat

Poslední důležitou činností je odstranění zvolených dat. Skládá se z:

```

DELETE FROM <tabulka>
WHERE      <sloupec> = '<hodnota>'

```

Opět je potřeba nezapomenout na klauzuli WHERE, bez které by zmizela všechna data v tabulce. Uvedeným příkladem bude smazání závodníka podle jeho ID:

```
DELETE FROM zavodnik WHERE WHERE id='TJN8901'
```

5.6 Manipulace s daty

Abychom mohli manipulovat s daty v databázi, nestačí nám jenom znát příkazy jazyka SQL. K tomu je zapotřebí vytvořit objekt třídy *SqlCeCommand*. Do konstruktoru předáme zmiňovaný SQL dotaz ve formě řetězce a instanci třídy *SqlCeConnection*. Pomocí metod obsahujících ve třídě provedeme požadované operace.

Příklad – načtení trati do databáze

```

string dotaz = "INSERT INTO Trat(tratID, delka, prevyseni, pocet_k)
              VALUES(H21, 13,2, 350, 26)";
SqlCeCommand cm = new SqlCeCommand(dotaz, cn);
cm.ExecuteNonQuery();

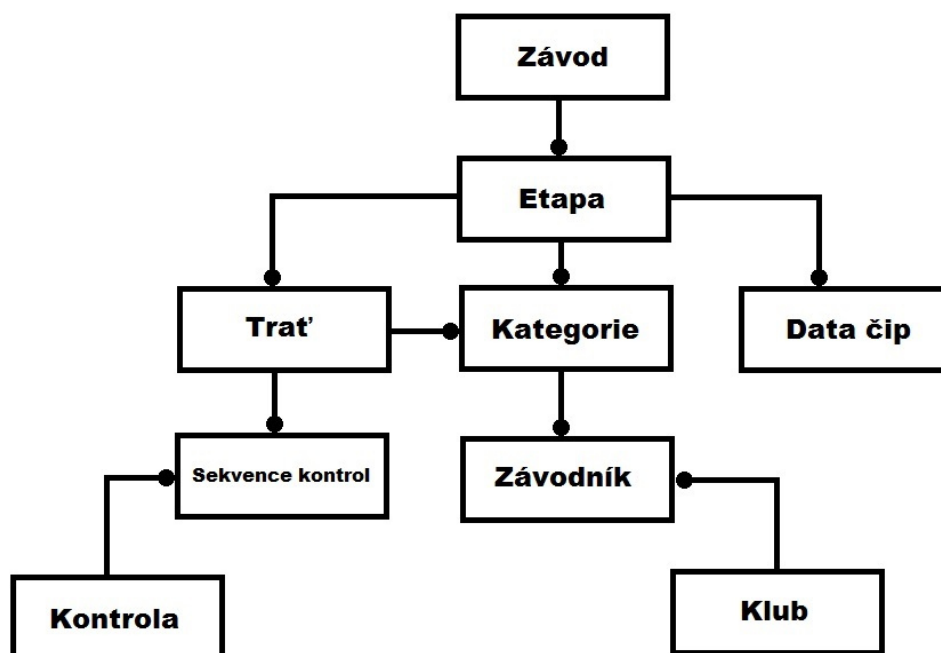
```

Metoda *ExecuteNonQuery* slouží ke spuštění jakéhokoliv příkazu, který nevrací žádnou datovou sadu z databáze. Tím jsou příkazy pro vkládání, úpravu či smazání záznamu. Pokud však chceme data číst, je zápis kódu odlišný:

```
using (SqlCeCommand com = new SqlCeCommand("SELECT tratID FROM Trat", cn))
{
    SqlCeDataReader reader = com.ExecuteReader();
    while (reader.Read())
        list.Add(reader.GetString(0));
}
```

K přečtení žádaných dat slouží metoda *ExecuteReader*, vracející objekt *SqlCeDataReader*. Následně je využita metoda *Read*, která vrací data po řádkách. V uvedeném příkladu požadují přečtení atributu ID trati ze všech existujících řádků. Data se načítají do kolekce list, dokud program neprojde všechny řádky.

5.7 Struktura navržené databáze



Obr. 5.7-1: Struktura databáze

Příložený model je složen z množství entit, které jsou mezi sebou propojeny a provázány vztahy. Vrchol modelu představuje entita závod, která může nabývat množstvím etap. Tato práce se bude zabývat „jen“ jedno-etapovými závody, vzhledem k jejímu rozsahu. Avšak model je připraven pro tvorbu více-etapových závodů. Každá etapa obsahuje různé typy tratí s určitou posloupností kontrol. V etapě jsou též k dispozici různé druhy kategorií, které musí mít přiřazenou určitou trať. Etapa taktéž

obsahuje množství přihlášených závodníků do závodu, reprezentující svůj klub. V poslední řadě etapa disponuje entitou data čip, kde jsou shromažďovány všechny informace ze závodního průkazu závodníka. Detailní popis databázových tabulek je přiložen v příloze.

6 SPUŠTĚNÍ APLIKACE

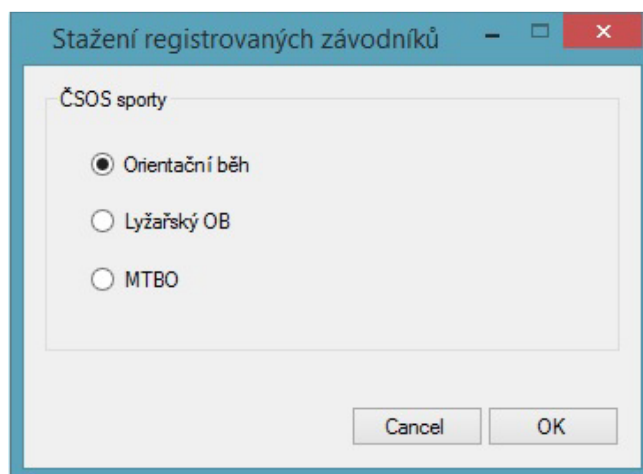
Jakmile je program spuštěn, zobrazí se uživateli dialogová správa o možnosti stažení souboru registrovaných závodníků. Každý registrovaný člen ve svazu má svoje unikátní ID resp. registrační číslo. Tvoří jej zkratka oddílu s následným čtyřčíslem, kde první dvě čísla znamenají ročník narození, další dvě jsou volitelné s výjimkou, že muži mají rozsah od 0 do 49 a ženy od 50 do 99.

Příklad: VIC8931 – muž, ročník 1989, oddíl OB Vizovice.

Stažený soubor obsahuje ke každému registračnímu číslu celou škálu údajů. Moje aplikace využije jen některé z nich a uloží je do textového souboru. Údaje jsou odděleny středníkem, celý blok údajů registrovaného člena je psán na každý řádek zvlášť

Příklad: VIC8931;Jan;Panovec;52349;C;VIC;OB Vizovice;

K čemu jsou tato data vhodná? Budeme-li chtít ručně přihlásit daného jedince do závodu, bude stačit zadat registrační číslo. Ostatní údaje budou doplněny automaticky. Pro každé odvětví v ČSOS jsou registrace závodníků evidovány samostatně. V klasickém OB může pod registrací VLI8901 evidován Jan Novák, naopak v LOB to může být např. Martin Poklop. Následně by mohlo dojít v programu k tomu, že budou vypsány nežádoucí údaje. Proto je zapotřebí specifikovat z jakého odvětví chceme údaje stáhnout a to samé platí při zakládání nového závodu.



Obr. 6-1: Dialogové okno s možností stažení registrovaných závodníků z vybraných odvětví

Generování závodníků z webové databáze ORIS trvá „poměrně dost času“, a proto je výhodné využít dalšího vlákna. Data tudíž budou stahována na pozadí programu.

```
WebClient webclient = new WebClient();
Timer      timer_TM = new Timer();

if (System.Net.NetworkInformation.NetworkInterface.GetIsNetworkAvailable())
{
    if (!Directory.Exists("data"))
        Directory.CreateDirectory("data");

    webclient.DownloadFileAsync(new Uri(adres), FilePath );
    timer_TM.Start();
}

else
    MessageBox.Show("Chyba v připojení k internetu",
                    "Upozornění", MessageBoxButtons.OK, MessageBoxIcon.Error);
```

Výše uvedený příklad popisuje, jak stáhnout ve vláknu určený soubor. K práci s internetem je využita třída *WebClient*. Ještě před startem stahování dat je nutno ošetřit, je-li dostupné připojení k internetu. V případě selhání program vypíše dialogovou zprávu o chybě. Všechny soubory jsou stahovány do složky data. Pokud složka neexistuje, je automaticky vytvořena pomocí metod třídy *Directory*. Následně metoda *DownloadFileAsync* spustí stahování souboru ve vláknu. Argumenty metody jsou webová adresa daného souboru a cesta k souboru, kam se budou ukládat data. Pokud v průběhu stahování dojde výpadku připojení k internetu, je vytvořen časový limit, ve kterém je nutno soubor stáhnout. Je-li překročen, stahování se přeručí funkcí *CancelAsync* a program opět vypíše zprávu o vypršení časového limitu. K vytvoření časového limitu je ze sady nástrojů použito třídy *Timer*. V ní nastavíme čas, za který se bude automaticky spouštět vytvořená metoda *timer_TL_Tick*.

```
private void timer_TL_Tick(object sender, EventArgs e)
{
    MessageBox.Show("limit pro spojení vypršel",
                    "Upozornění", MessageBoxButtons.OK, MessageBoxIcon.Error);





    timer_TM.Stop();
    webclient.CancelAsync();
}
```

7 ZALOŽENÍ NOVÉHO ZÁVODU

Po spuštění aplikace má uživatel na výběr pouze založit novou událost nebo ji otevřít. Nyní se budeme zabývat první možností. Zvolením vybraného odvětví události je otevřeno dialogové okno složené ze dvou částí. V první části je vyžadováno vyplnění základních informací o závodě. Druhá část se zabývá načtením závodníků.

7.1 Základní informace

Na obr. 7.1-1 jsou vidět parametry, které musí uživatel bezpodmínečně vyplnit. V programu je ošetřeno, že uživatel nesmí použít znak „+“ v názvu závodu. Důvod je takový, že v atributech primárních klíčů je využíváno tohoto znaménka a tudíž by docházelo ke kolizím v programu. Práce s formulářovými prvky ze sady nástrojů ve Visual studiu je velmi jednoduchá a ušetří poměrně dost času.

Název závodu:	<input type="text" value="Ocad cup"/>
Místo závodu:	<input type="text" value="Liberec- Harcov"/>
Organizátor:	<input type="text" value="Ocad team"/>
Datum:	<input type="text" value="7. 3. 2015"/> 
Čas startu:	<input type="text" value="10:00"/> 
Start. interval(min):	<input type="text" value="2"/> 
Časový limit(min):	<input type="text" value="90"/> 

Obr. 7.1-1: Dialogové okno s povinným vyplněním základních informací o závodě

7.2 Načtení závodníků

Druhá část formuláře se zabývá nahráním informací o závodnících do databáze. Uživatel má k dispozici tři varianty, jak načíst data. První možností je využití webového systému ORIS. Další variantou je načtení dat z textového souboru. Poslední způsob je ruční načtení. Jednotlivé varianty si na následujících řádcích podrobně vysvětlíme.

7.2.1 Načtení dat ze systému ORIS

Není nic pohodlnějšího, než když člověk klikne na tlačítko a na pozadí toho se provede mnoho užitečných operací. Toto lze s lehkou nadsázkou říct i o stisknutí tlačítka „stáhnout“ v této aplikaci. Kliknutím se stáhnou všechna potřebná data, upraví se do určité formy a jsou připraveny k načtení do databáze. Všechny události nahrané na systému ORIS jsou možné k nahlédnutí i v této aplikaci. Uživatel v podstatě ani nemusí znát webovou adresu systému ORIS. Důležité však je, aby měl k dispozici internetové připojení.

The screenshot shows a web application interface for loading data from the ORIS system. It features a section titled 'Typ načtení' (Type of loading) with three radio button options: 'Načtení závodníků z ORISU' (selected), 'Načtení závodníků ze souboru' (unselected), and 'Ruční načtení' (unselected). Below this is a section titled 'Načtení závodníků z ORISU' (Loading athletes from ORISU). It includes a dropdown menu for the year, currently set to '2015', and another dropdown for the month, set to 'Leden' (January). A list of races is displayed below, with the following items visible: '10.01.2015 - 4. přeshraniční skirogaining 2014 - LIV', '10.01.2015 - Opět na Kozla - KAM', '11.01.2015 - Městský sprint Pardubice - LPU', '17.01.2015 - Radostický sprint - BBM', '17.01.2015 - 6. závod Zimní ligy Hanácké oblasti - OOL', '23.01.2015 - 5. závod Slezské zimní ligy - HAV', '24.01.2015 - Po stopách císařovny Elišky - SCP', '30.01.2015 - Mosty PZL - TAP', '31.01.2015 - Hromniční Trápení 2015 - RBK', '31.01.2015 - Mistrovství ČR na klasické trati - SZP', and '31.01.2015 - Český pohár, ŽA, ŽB (noční sprint) - SZP'.

Obr. 7.2.1-1: Zobrazený seznam závodů na měsíc leden roku 2015

Uživateli se zobrazuje seznam závodů zvoleného měsíce a příslušného roku. Všechny tyto závody jsou zjištěny ze zdrojového kódu webové stránky. Proto je potřeba zdrojový kód stáhnout a nezbytné informace si v něm vyhledat.

Příklad: Zjištění všech roků, kde se vyskytují závodní akce

```
<div class="dp25">
  "
  &nbsp;"
  <a class="gLink" href="#" p1="2013-01-01" p2="2013-12-31">2013</a>
  "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"
  <a class="gLink" href="#" p1="2014-01-01" p2="2014-12-31">2014</a>
  "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"
  <a class="gLink tb" href="#" p1="2015-01-01" p2="2015-12-31">2015</a>
  "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"
  <a class="gLink" href="#" p1="2016-01-01" p2="2016-12-31">2016</a>
  "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"
  "
</div>
```

Obr. 7.2.1-2: Část zdrojového kódu z adresy <http://oris.orientacnisporty.cz>

V prvním kroku je stažen zmíněný zdrojový kód webové adresy, jenž je následně uložen do řetězce např. názvu „s“. V řetězci se vyhledá úsek, kde se vyskytují požadované informace, viz obrázek výše. Úsek je získán pomocí dvou unikátních podřetězců, které ho ohraničují. Pomocí prvního z nich dostaneme startovní index úseku a pomocí druhého index cílový.

```
string s1 = "div class=\"dp25\">";
string s2 = "</div>";
int sIndex = s.IndexOf("div class=\"dp25\">") + s1.Length;
int eIndex = s.IndexOf(s2, sIndex);

s = s.Substring(sIndex, eIndex - sIndex);
```

Ze vzniklého úseku je však potřeba opakovaně použít tento systém hledání, dokud nejsou získány již jen požadované podřetězce. Níže je zobrazen kód s načtením existujících roků do položek kontrolky ComboBox (pole výběru). Funkce *ExtractString* vrací vyžadující část vstupující řetězce.

```
string web_str = "";
WebClient wc = new WebClient();

web_str = wc.DownloadString("http://oris.orientacnisporty.cz");
web_str = ExtractString(web_str, "div class=\"dp25\">", "</div>");

while( web_str.Contains("</a>"))
{
    c_roky.Items.Add(ExtractString(web_str, ">", "</a>"));
    web_str = web_str.Substring(web_str.IndexOf("</a>", 0) + 4);
}
```

Obdobně jako předchozí příklad jsou vyhledány informace o názvu závodu a taktéž kód události, který je nutný specifikovat, při stažení dat vybrané akce.

Příklad: Řetězec s odkazem závodu, který je uložen pod kódem 2046

```
string odkaz = "http://oris.orientacnisporty.cz/ExportPrihlasek?mode=oe2010&id=" + "2046";
```

Pokud stažení zvoleného závodu proběhne bez problému, zobrazí se nápis „OK“, v opačném případě se zobrazí nápis „CHYBA“. V nabídce se mohou vyskytovat štafetové závody, které mají odlišný formát exportu dat a také nejsou žádoucí pro program, protože s tímto druhem závodění program pracovat neumí. Je proto potřeba zjistit, kdy se jedná o nežádoucí událost. V žádoucích událostech je každý závodník psán na samotný řádek a může nabývat x různých údajů oddělené středníkem, tudíž postačí ošetřit počet x středníků na každém řádku. Chce-li uživatel změnit zvolený závod, pak postačí stisknout tlačítko „zrušit“ a program předchozí stažená data vymaže.

7.2.2 Načtení dat z textového souboru

Klasický způsob, využívaný hlavně v jiných programech, je načítání dat ze souboru. Může se stát, že závod není registrovaný na systému ORIS nebo není dostupné připojení k internetu, a tak tato možnost načítání dat může přijít vhod.

The screenshot shows a dialog box titled 'Typ načtení' (Type of loading) with three radio button options: 'Načtení závodníků z ORISU', 'Načtení závodníků ze souboru' (selected), and 'Ruční načtení'. Below this is a section titled 'Načtení závodníků ze souboru' (Loading racers from file) containing two rows of controls. The first row is for 'Závodníci' (Racers) and the second for 'Vklady:' (Entries). Each row has 'načíst' (load) and 'zrušit' (cancel) buttons, followed by 'OK' and a help icon '?'.

Obr. 7.2.2-1: Zobrazený typ načítání ze souboru

Program vyžaduje dva soubory. V prvním jde o data o závodníkovi, ve druhém o výše vkladů vypsanych kategorií. Formát zápisu dat je obdobný exportovaným datům z ORISU. Uživatel si může předepsaný formát dat prohlédnout stiskem tlačítka „?“.

Formát pro soubor se závodníky:

"Reg._číslo;Kategorie;Čip;Příjmení;Jméno;Licence;Zkratka_oddílu;Termín;Oddíl;Komentář;
Formát pro soubor se vklady:

Kategorie;Termín_1;Termín_2;Termín_3;

Ošetření souborů je oproti datům z ORISu trochu složitější. Data o závodníkovi či o vkladech kategorií se opět píše na každý řádek zvlášť. Každý řádek musí obsahovat předem daný počet středníků. Chybí-li nějaký parametr, napíše se prázdné místo, avšak středník nesmí chybět.

Příklad: Příklad ošetření počtu středníků na řádku souboru se vklady

```
pocet_s = 4;
using (StreamReader sr = new StreamReader(openFileDialog.OpenFile(), Encoding.Default))
{
    while (!sr.EndOfStream)
    {
        string s = sr.ReadLine();
        if (s.Count(x => x == ';') != pocet_s)
        {
            nacti_status(label, false, zrus, nacti);
            return;
        }
    }
}
```

Program čte soubor po jednotlivých řádcích. V podmínce se porovnává počet středníků s předem daným počtem. V případě neúspěchu spustí funkci, která vypíše chybu.

Další kontrolou je, jsou-li vyplněny parametry u registračního čísla, kategorie a oddílu. Ty jsou totiž primárními klíči příslušných entit, a proto nesmí chybět. Navíc každé registrační číslo musí být unikátem, musí obsahovat minimálně 7 znaků dle pravidel ČSOS a nesmí začínat předponou „VAK“, určené pro vakanty. Dále je testováno číslo čipu, termín a vklady v termínech, které mohou obsahovat pouze číselné znaky. K tomu termín může být v rozmezí intervalu <1;3>. Posledním testem je porovnání kategorií v obou souborech. Přesněji řečeno, každá přiřazená kategorie u závodníka musí být obsažena v souboru se vklady.

7.2.3 Ruční načtení

Nemáme-li k dispozici data z ORISU nebo soubor s daty, je možné načíst data ručně. Samotné načítání je možné provést až v okně s úpravou záznamů, které bude vysvětleno později.

7.3 Dokončení a nahrání do databáze

Jsou-li vyplněné všechny základní informace a byly provedeny příslušné kroky k načtení závodníků, jsou data připravena k dokončení a nahrání do databáze. Po potvrzení tlačítkem „OK“ systém ověří, zda byly všechny položky vyplněny. Pokud ne, objeví se u nevyplněných položek varovný nápis. Jestliže je všechno v pořádku, program vytvoří složku odpovídající závodu, zkopíruje výchozí šablonu databázového souboru a

začne do něj nahrávat náležitá data. Program také kontroluje, jestli už nebyl obdobný soubor vytvořen.

Příklad: kopírování šablonového souboru do nového:

```
string fileName = "databaze.sdf"; // název šablony
string fileName2 = "databaze_" + Navez_z_box.Text + ".sdf"; // název nového souboru
//cesty, odkud se má soubor vzít a nahrát, pokud je soubor v kořenové složce programu, není
//potřeba cestu specifikovat
string sourcePath = "";
string targetPath = "data\\race" + "_" + Navez_z_box.Text + "_" + datum_zav.Text + "\\";

//spojení řetězce cesty s názvem souboru
string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
string destFile = System.IO.Path.Combine(targetPath, fileName2);
//metoda pro kopírování souboru
System.IO.File.Copy(sourceFile, destFile, true);
```

V závěru je otevřen textový soubor s informacemi o vytvořeném závodě. Krom uživatelem vyplněných informací, jsou vypsány počty importovaných závodníků, oddílů a kategorií. Dalšími položkami jsou varovné zprávy týkající se duplikovaných čísel startovního průkazu. V samotném závodě, je však možné použít určitý čip jen jednou. Dále jsou vypsány duplikace jmen závodníků.

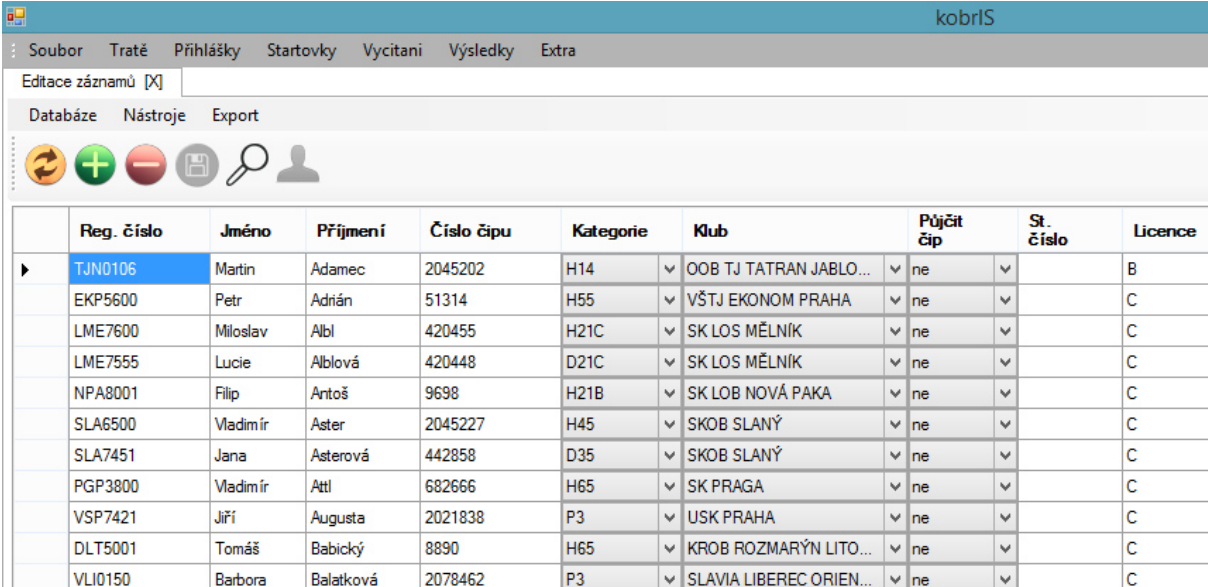
```
VÝSLEDKY:
-----
NÁZEV:           Jarní skály
MÍSTO:           Kost
ORGANIZÁTOR:     TUV
DATUM:           11. 3. 2015
START:           10:00
STARTOVNÍ INTERVAL 2
ČASOVÝ LIMIT     90
POCET ETAP:     1
-----
Počet importovaných závodníků: 922
Počet importovaných oddílů: 102
Počet importovaných kategorií: 23
-----
VAROVÁNÍ!!!
ZÁVODNÍCI duplikát čísla čipu:
<2053614>
<2078464>

ZÁVODNÍCI duplikát jména:
<Jaromír Pospíšil>
<Lenka Svobodová>
```

Obr. 7.3-1: Náhled textové souboru s informacemi o založeném závodě

8 EDITACE ZÁZNAMŮ

Velmi důležitou činností pořadatele závodu, ačkoliv velmi nezáživnou, je kontrola a příprava důležitých dat pro závod. Pod položkou editace záznamů je možno nalézt všechna data o závodnících, kategoriích a klubech. Uživatel má k dispozici množství funkcí, jak příslušné parametry i konkrétní data měnit. Všechny funkce budou následně podrobně vysvětleny. Data jsou uspořádána do tabulky podobně jako u Excelu pomocí komponenty *DataGridView*. Sloupce obsahují jednotlivé parametry záznamů, které jsou psány do samostatných řádků.



	Reg. číslo	Jméno	Příjmení	Číslo čipu	Kategorie	Klub	Půjčit čip	St. číslo	Licence
▶	TJN0106	Martin	Adamec	2045202	H14	OOB TJ TATRAN JABLO...	ne		B
	EKP5600	Petr	Adrián	51314	H55	VŠTJ EKONOM PRAHA	ne		C
	LME7600	Miloslav	Albi	420455	H21C	SK LOS MĚLNÍK	ne		C
	LME7555	Lucie	Albová	420448	D21C	SK LOS MĚLNÍK	ne		C
	NPA8001	Filip	Antoš	9698	H21B	SK LOB NOVÁ PAKA	ne		C
	SLA6500	Vladimír	Aster	2045227	H45	SKOB SLANÝ	ne		C
	SLA7451	Jana	Asterová	442858	D35	SKOB SLANÝ	ne		C
	PGP3800	Vladimír	Atti	682666	H65	SK PRAHA	ne		C
	VSP7421	Jiří	Augusta	2021838	P3	USK PRAHA	ne		C
	DLT5001	Tomáš	Babický	8890	H65	KROB ROZMARÝN LITO...	ne		C
	VLI0150	Barbora	Balatková	2078462	P3	SLAVIA LIBEREC ORIEN...	ne		C

Obr. 8-1: Náhled okna s editováním záznamů

8.1 Změna parametru - buňky

Podobně jako v Excelu lze libovolně editovat každou buňku. Stačí se dostat na požadovaný parametr, ať už pomocí myši nebo klávesnice. Buňky jsou buď typu „TextBox“ – volně editovatelné pole nebo typ „ComboBox“ – výběr položky ze seznamu. Je-li hodnota změněna, má uživatel možnost stisknout tlačítko „uložit“ nebo postačí prosté opuštění buňky. Program spustí dialogovou zprávu s otázkou, má-li být parametr opravdu změněn.

Program si zjistí, jestli záznam opravdu existuje, protože pracuje-li s databází více uživatelů v jedné době, tabulka nemusí být aktuální. Pokud tedy záznam neexistuje, program se zeptá, jestli má být uložen celý řádek, nejen editovaný parametr. Pro buňky s typem „TextBox“- registrační číslo, kategorie a oddíl je ověřeno, jestli buňka při uložení není prázdná, neobsahuje-li znak „+“ nebo není-li duplikátem. Navíc registrační číslo nesmí začínat příponou „VAK“ určenou pro vakanty. Taktéž musí obsahovat minimálně 7 znaků, dáno předpisy ČSOS. Kontrola zápisu číselných hodnot probíhá u buněk: číslo čipu, startovní číslo, startovní interval, časový limit, vklady, počet vakantů, půjčení čipů, startovné závodníci, ostatní služby, celková cena, zaplacení. U některých z nich je také provedena kontrola proti uložení prázdného pole. Posledním testováním je správný formát hodin (HH:MM), kde musí být splněno:

1. Řetězec obsahuje přesně 5 znaků
2. první a poslední dva znaky v řetězci jsou číselná hodnota
3. třetí znak je symbol „:“
4. číselná hodnota prvních dvou znaků je menší než 23
5. číselná hodnota posledních dvou znaků je menší než 59

Test prázdné buňky:

```
if (database.Rows[row].Cells[col].Value == null)
```

Test řetězce, jestli obsahuje znak „+“:

```
if (database.Rows[row].Cells[col].Value.ToString().Contains('+'))
```

Test délky řetězce:

```
if (database.Rows[row].Cells[col].Value.ToString().Count() < 7)
```

Test řetězce obsahující slovo „VAK“ na začátku řetězce :

```
if ( database.Rows[row].Cells[col].Value.ToString().Substring(0, 3) == "VAK")
```

Test duplikátu konkrétního sloupce:

```
int r = 0;
foreach (DataGridViewRow item in database.Rows)
{
    if (row != r &&
        item.Cells[0].Value.ToString()==database.Rows[row].Cells[col].Value.ToString())
        break;

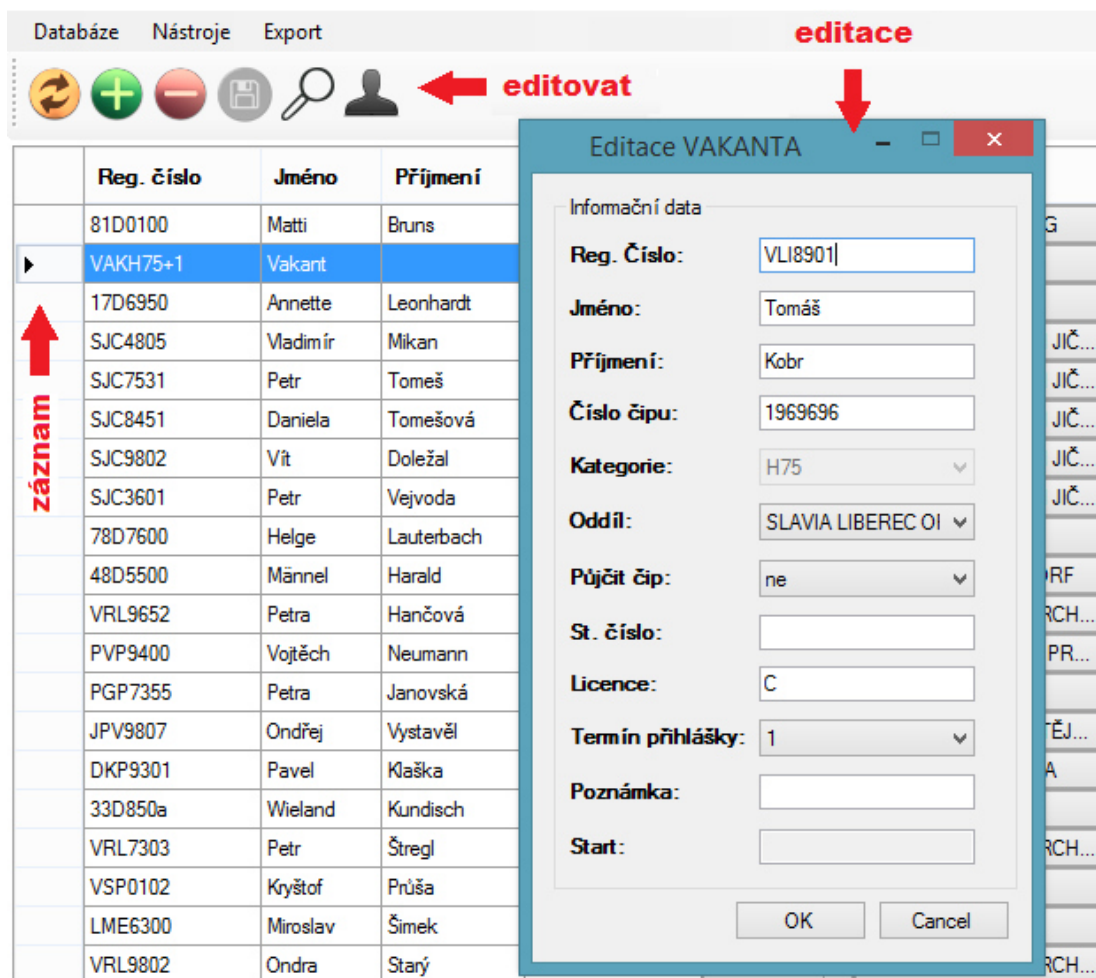
    r++;
}
```

Test znaku na konkrétní pozici:

```
if ( database.Rows[row].Cells[col].Value.ToString()[2] != ':' )
```

8.2 Editace záznamu VAKANT

V editaci záznamů konkrétně v sekci kategorie najdeme sloupec s názvem počet vakantů. Počet lze mít pro každou kategorii jiný. Zvolením určitého počtu dochází k automatickému vytvoření a načtení do databáze. Rozhodne-li se pořadatel změnit jejich počet v dané kategorii, stačí přepsat hodnotu a program automaticky přidá nebo ubere dle difference mezi původním a žádaným počtem.



Obr. 8.2-1: Editování vakanta

V sekci závodníci jsou záznamy vakantů charakteristické registračním číslem složené z předpony „VAK“ s následným názvem kategorie. Slovo pokračuje symbolem „+“ a číslem určující pořadí vakanta v kategorii. Zvolením celého řádku se objeví tlačítko pro editování záznamu a jeho následným stisknutím se otevře dialogové okno pro editaci. Nyní konečně bude moci být využito staženého souboru registrovaných závodníků. Konkrétní ID závodníka, kterého chceme přihlásit, vypíšeme do kolonky

registrační číslo. Ostatní parametry budou doplněny automaticky. Program ověří, jestli je vytvořen oddíl závodníka. Není-li vytvořen, spustí se dialogová zpráva s nabídkou automatického vytvoření.

Příklad: Automatické doplnění parametrů registrovaného závodníka

```
//vyhledání závodníka v seznamu
int index = list_reg.FindIndex(x => x.f_reg == reg_id.Text);

//ověření, existuje-li závodník
if (index !=-1)
{
    jmeno.Text = list_reg[index].f_jm;
    prijmeni.Text = list_reg[index].f_pr;
    c_cipu.Text = list_reg[index].f_cc;
    licence.Text = list_reg[index].f_lic;

    //ověření, existuje-li oddíl
    if (odd.Items.Contains(list_reg[index].f_odd))
        odd.Text = list_reg[index].f_odd;
    else
    {
        DialogResult result = MessageBox.Show("Oddíl nalezeného závodníka není vytvořen,
        chcete jej vytvořit?", "Upozornění",
        MessageBoxButtons.YesNo,MessageBoxIcon.Question);

        if (result == System.Windows.Forms.DialogResult.Yes)
        {
            klub kl = new klub(connectionstring);

            //nahrání klubu do databáze
            using (SqlCeConnection cn = new SqlCeConnection(connectionstring))
            {
                kl.openDatabase(cn);
                kl.add_data(list_reg[index].f_odd.ToUpper(), list_reg[index].f_zo, "0", "0", cn);
                cn.Close();
            }

            odd.Items.Add(list_reg[index].f_odd);
            odd.Text = list_reg[index].f_odd;
        }
    }
}
```


8.3 Přidání záznamu

Ruční přidání záznamu vytvoříme pomocí tlačítka „PLUS“. Program vygeneruje nový řádek v komponentě *DataGridView* a skočí do první buňky. Zde uživatel vyplní potřebná data. Opět funguje automatické načtení závodníka dle registračního čísla, které bylo popsáno v předchozí kapitole.

8.4 Odebrání záznamu

Chceme-li určitý záznam vymazat, použijeme tlačítka „MINUS“. Abychom mohli daný záznam vymazat, je potřeba označit celý řádek. Program umožňuje i mazání více záznamů najednou. Je-li smazána určitá kategorie, jsou díky relaci mezi entitami smazáni i všichni závodníci načtení v této kategorii. Stejným stylem funguje odstranění žádaného oddílu.

Příklad: Smazání vybraných oddílů v komponentě *DataGridView*

```
klub k1 = new klub(connectionstring);

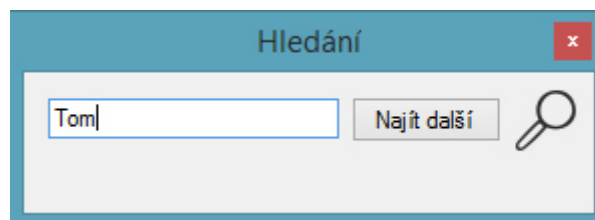
using (SqlCeConnection cn = new SqlCeConnection(connectionstring))
{
    k1.openDatabase(cn);
    foreach (DataGridViewRow item in database.SelectedRows)
    {
        //oddíl VAKANT nelze smazat
        if (item.Cells["oddil"].Value.ToString() != "VAKANT")
            k1.delete_row(cn, Cells["oddil"].Value.ToString());
    }
    cn.Close();
}
```

8.5 Aktualizace záznamů

Používáním databáze z více počítačů v jednom čase mohou být data měněna každou chvíli. Proto je potřeba občas záznamy aktualizovat. Tu lze obnovit pomocí přiloženého tlačítka. Obnova všech záznamů taktéž probíhá po každé uživatelské změně. Je-li změněn pouhý jeden parametr, tabulka se načte kompletně aktuální.

8.6 Vyhledávání parametru 🔍

Závodů v orientačním běhu se často zúčastňuje mnoho lidí (řádově stovky) a hledání parametrů v záznamech by se mohlo stát zlým snem. Pro usnadnění byla vytvořena funkce hledání parametrů. Stisknutím příslušného tlačítka se objeví dialogové okno, kde uživatel vypíše hledaný výraz a program najde první v pořadí. Opětovným stisknutím se pokusí najít další. Jakmile dojde na konec tabulky, oznámí uživateli, že je na konci a přejde opět na začátek tabulky. Jestliže hledaný výraz není nalezen, zobrazí se chybová hláška. K nalezení hledaného slova stačí zadat též pouze jeho část, nevádí ani záměna velkých a malých písmen.



Obr. 8.6-1: Dialogové okno pro hledání výrazu

8.7 Změna závodníka na vakanta

Může se stát, že např. závodník onemocní a tak pořadateli závodu zbude volné místo v kategorii. Tato funkce snadno převede závodní pozici na vakanta.

8.8 Spojení kategorií

Jakmile skončí termín pro přihlášení do závodu, pořadatel zná zhruba počty závodníků v jednotlivých kategoriích. Má právo, v případě malého počtu závodníků, sloučit více kategorií dohromady. Uživatel v programu označí řádky kategorií, které chce sloučit a spustí danou funkci. Spustí se dialogové okno vyzývající uživatele ke zvolení kategorie, kterou chce z vybraných zachovat spolu s jejími parametry. Po dokončení se účastníci z okolních kategorií do zvolené přesunou.

8.9 Rozdělení kategorií

Opačným případem spojení kategorií je její rozdělení. Často se stane, především v hlavní kategorii, že počet přihlášených účastníků do dané kategorie je velký. Je-li zvolen intervalový start, tak při počtu 100 lidí v kategorii s intervalem 3 minuty, bude čas odstartování celé kategorie trvat 300 minut resp. 5 hodin. Z tohoto hlediska to není vždycky žádoucí, proto byla zavedena tato funkce.

Obr. 8.9-1: Dialogové okno pro rozdělení kategorií

V dialogovém okně uživatel před samotným rozdělením vyplní požadované informace. Zvolením kategorie se zobrazí přehled o jejím počtu přihlášených. Počtem podkategorií se rozumí, na kolik skupin se má kategorie rozdělit. Do těchto skupin je potřeba zvolit počet závodníků. Vytvoří-li např. uživatel 2 skupiny a do první přidá 30 závodníků z původních 68 osob, není potřeba do druhé již nic zadávat, automaticky se do ní načte zbylých 38 osob. Názvy podkategorií budou obsahovat název předešlé

kategorie s rozlišovacím symbolem formou číslice nebo znaku. Rozdělení osob může proběhnout podle několika kritérií:

RANDOM:

Jde o náhodné rozdělení, pro které je využita třída *Random*.

Příklad: načtení do skupin, dle náhodného rozdělení:

```
public List<podkategorie> l_pkat = new List<podkategorie>();
Random r = new Random();

//přečtení závodníků ze seznamu
for (int i = 0; i < l_pkat.Count; i++)
{
    //načtení počtu závodníků do poslední skupiny
    if ((i + 1) == l_pkat.Count && Int32.Parse(l_pkat[i].pocet_z) < zav.list.Count)
        l_pkat[i].pocet_z = zav.list.Count.ToString();

    //cyklus probíhající do zvoleného počtu závodníků ve skupině nebo dokud existují
    // závodníci
    for (int j = 0; j < Int32.Parse(l_pkat[i].pocet_z); j++)
    {
        if (zav.list.Count > 0)
        {
            //generování náhodného čísla
            int k = r.Next(0, zav.list.Count);
            //vypsání závodníka do skupiny
            l_pkat[i].box_zav.Add(zav.list[k].reg_cislo.ToString()+
                                zav.list[k].jmeno.ToString() +
                                " " + zav.list[k].prijmeni);

            //vymazání vygenerovaného závodníka ze seznamu
            zav.list.RemoveAt(k);
        }
        else
            break;
    }
}
```

RANKING:

Speciální název pro dlouhodobý žebříček závodníků v hlavní kategorii dospělých je nazýván jako ranking. Existují dva druhy - ženský a mužský. Do tohoto žebříčku se počítá 10 nejlepších výkonů za uplynulých 24 měsíců příslušného závodníka. Pořadí v rankingu je aktualizováno koncem každého měsíce. Všechny tyto měsíční pořadí jsou opět z ORISU do programu nahrány. Lze tedy jednoduše zadat měsíc, podle kterého se stáhne žádaný ranking. Následně se z tohoto pořadí vytvoří skupiny. Pokud přihlášený závodník není v žebříčku nalezen, řadí se za posledního evidovaného závodníka.

1	Šedivý	Jan	PGP8409	101718	10000	1
2	Nykodým	Miloš	ZBM9005	101474	9976	2
3	Procházka	Jan	PGP8407	100454	9876	3
4	Král	Vojtěch	SSU8841	98029	9637	4
5	Chromý	Adam	ZBM8841	97274	9563	5
6	Hájek	Daniel	ZBM8919	96127	9450	6
7	Kubelka	Tomáš	LPU9307	96064	9444	7
8	Kubát	Pavel	PHK9101	96013	9439	8
9	Chloupek	Adam	ZBM9300	95512	9390	9
10	Semík	Ondřej	KAM9401	95285	9368	10
11	Procházka	David	LPU9113	95107	9350	11
12	Hradec	Pavel	TUR8801	94827	9323	12
13	Losman	Petr	PHK7901	94705	9311	13
14	Poklop	Martin	SSU8802	94565	9297	14
15	Udržal	Tomáš	LPU8002	94459	9286	15
16	Mrázek	Jan	SJC8102	94203	9261	16
17	Bravený	Vít	ZBM9102	93871	9229	17
18	Schuster	Marek	AOP9303	93784	9220	18
19	Kamenický	Jakub	PHK9104	93687	9210	19
20	Minář	Marek	ASU9444	93552	9197	20

Obr. 8.9-2: Náhled pořadí mužského rankingu

Příklad: načtení pozic z rankingu

```
//nahrání pozic klasifikovaných lidí

Using (StreamReader sr = new StreamReader("data\\" + nazev + ".txt", Encoding.Default))
{
    string s;
    //čtení řádků v souboru

    while ((s = sr.ReadLine()) != null)
    {
        for (int i = 0; i < zav.list.Count; i++)
        {
            string reg_cislo = zav.list[i].reg_cislo;
            string s2= s.Substring(28, 7);

            //je-li nalezena klasifikovaná osoba, přiřaď pozici
            if (s2 == reg_cislo)
                zav.list[i].pozice = s.Substring(0, 5).Trim();
        }
    }
}

//nahrání pozic neklasifikovaných lidí
for (int i = 0; i < zav.list.Count; i++)
    if (zav.list[i].pozice == null)
        zav.list[i].pozice = "99999";

// třídění podle pořadí
zav.list = zav.list.OrderBy(x => Int32.Parse(x.pozice)).ToList();
```

ŽEBŘÍČEK A:

Alternativa rankingu je tzv. žebříček A, který se týká dorosteneckých a juniorských kategorií. Rozdělení do skupin probíhá obdobně jako v předešlém případě.

8.10 Rozdělení do kvalifikačních skupin

Speciálním případem rozdělení kategorií, jsou kvalifikace. Kvalifikační skupiny se vytvářejí pro Mistrovství republiky v individuálních disciplínách resp. pro dorostenecké, juniorské a hlavní dospělé kategorie. Tyto skupiny dané kategorie startují paralelně ve společných minutách. Závodník se kvalifikační skupinu dozví až v posledním koridoru v předstartovní zóně. Díky tomu vzniká anonymita, což je výhoda. Dejme tomu, že jsou v kategorii vytvořeny čtyři kvalifikační skupiny, takže závodníci startují po čtyřech osobách v minutě. Interval za nimi tvoří další čtyři, kteří ale netuší, jaké je rozložení do skupin závodníků před nimi. Jediné co vědí, že nemají společnou kvalifikační skupinu se závodníky ve stejné minutě. Další výhodou kvalifikačních skupin je snížení času pro odstartování celé kategorie.

The screenshot shows a software interface for creating qualification groups. It features two baskets, '1. koš' and '2. koš', each containing a list of athletes with their IDs and names. A central 'NALOSOVAT' button is used to move athletes between baskets. 'Přesunout ->' and '<- Přesunout' buttons are also present for moving athletes from one basket to the other.

1. koš		2. koš
ZBM9005 Miloš Nykodým	Přesunout ->	VLI7700 Michal Horáček
KAM9401 Ondřej Semík		PGP7601 Luboš Matějů
LPU9113 David Procházka	NALOSOVAT	SJC8703 Jan Beneš
PHK7901 Petr Losman		TUR8103 Petr Fodor
SJC8102 Jan Mrázek	<- Přesunout	TUR8303 Tomáš Dlabaja
		TUR9502 Patrik Horák
		TUR8700 Martin Luštický

Obr. 8.10-1: pohled na losovací koše

V první řadě je potřeba vytvořit výkonnostní losovací koše podle žebříčků. Pro první dvě zmíněné kategorie je to opět „žebříček A“ a pro dospělé „Ranking“. Co se týče předpisů pro losování z hlediska disciplín, jsou s výjimkou sprintu pro dospělé stejné. Program umí prozatím losovat všechny kromě onoho sprintu. Jako příklad uvedu ve zkratce kritéria pro dospělou kategorii v disciplíně krátká a klasická trať. Do hlavní kategorie může postoupit prvních 200 lidí v pořadí dle rankingu (omezené věkem). Mimo ně mají právo startovat vítězové oblastního mistrovství v příslušné disciplíně a taktéž členové reprezentačního družstva a cizinci. Do prvního losovacího koše se řadí osoby do 20. místa v rankingu a členové reprezentačního družstva. Do druhého koše

jsou to závodníci v pořadí mezi 20-50. místem v rankingu, v třetím osoby mezi 50-90. místem, do čtvrtého ostatní závodníci do 200tého místa. V posledním koši jsou evidováni cizinci a vítězové oblastního mistrovství. Ostatní přihlášené bez práva startu program uloží do skupiny ostatní. Přiřazené závodníky lze pak ručně přesouvat mezi koši navzájem.

Program umí plně načíst závodníky do skupin z rankingu a rozpoznat osoby z cizích států. Avšak oblastní mistry a členy reprezentačního výběru je třeba doplnit ručně. Povedlo se mi načíst osoby z reprezentace z webu reprezentačního družstva, ale nastal problém, protože ke konkrétním jménům nebyly k dispozici registrační čísla. Poté docházelo ke kolizi, kdy se za reprezentanta „Jana Procházku“ nominoval do prvního koše jeho jmenovec, který toto právo neměl.

Příklad: Funkce pro naplnění závodníků do příslušného losovacího koše

```
private void losovani_do_kosu( zavodnik zav, int odkud, int kam, string kos)
{
    podkategorie p = new podkategorie();
    p.pod_kat = kos;

    // procházení všech závodníků
    for(int i = 0; i < zav.list.Count; i++)

        if (Int32.Parse(zav.list[i].pozice) > odkud && Int32.Parse(zav.list[i].pozice) <= kam)
        {
            p.box_zav.Add(zav.list[i].reg_cislo + " " +
                zav.list[i].jmeno + " " +
                zav.list[i].prijmeni);
        }

    p.pocet_z = p.box_zav.Count.ToString();
    l_kose.Add(p);
}
```

Jakmile jsou koše připraveny, postačí stisknout tlačítko losovat. Do vytvořených skupin jsou závodníci losováni náhodně po jednotlivých koších. Skupinu bez nároku na absolvování trati program vybědne k přiřazení k jiné (veřejné) kategorii, nebo pro ni vytvoří novou

9 TRATĚ

Tratě jsou prezentovány jako celek určité posloupnosti kontrol. Mohou být použity ve více kategoriích najednou. K načtení tratí lze v aplikaci využít volbu nahrání ze souboru anebo ruční načtení. Importované tratě je možné editovat nebo je vymazat. Grafická část tvorby tratí se vytvořena v různých programech. Nejznámější a nejpoužívanější program této tvorby je program *Ocad*. Tento program nabízí export dat vytvořené trati do textového souboru. Exportovaný soubor obsahuje údaje o názvu trati, její délce, převýšení a posloupnosti kontrol.

D10C	1,2	0	11	S1-51-55-58-56-59-48-49-63-64-46-100-F1
D12C	1,3	0	11	S1-54-53-31-51-58-56-44-48-49-63-100-F1
D14C	1,5	0	13	S1-60-32-55-51-56-44-43-48-49-45-64-46-100-F1
D16C	1,6	5	12	S1-61-33-54-55-32-41-42-43-44-47-48-100-F1
D18C	2,0	23	14	S1-61-54-43-42-41-40-50-35-56-44-52-46-45-100-F1
D20C	2,2	50	16	S1-53-50-39-38-37-36-35-41-42-59-57-52-46-64-45-100-F1
D21C	2,2	50	16	S1-53-50-39-38-37-36-35-41-42-59-57-52-46-64-45-100-F1
D35C	2,0	23	14	S1-61-54-43-42-41-40-50-35-56-44-52-46-45-100-F1
D45C	1,7	13	17	S1-39-35-50-33-61-31-55-54-43-44-52-46-64-45-63-49-100-F1
D55C	1,4	0	11	S1-54-43-44-56-58-62-47-52-48-49-100-F1
H10C	1,3	0	11	S1-51-55-62-59-57-47-48-63-64-46-100-F1
H12C	1,4	0	11	S1-54-33-31-51-56-57-59-48-49-45-100-F1
H14C	1,6	5	12	S1-61-33-54-55-32-41-42-43-44-47-48-100-F1
H16C	1,7	48	14	S1-38-37-36-35-39-50-33-58-59-47-46-45-49-100-F1
H18C	2,4	45	15	S1-60-32-54-42-40-50-34-36-37-38-44-52-46-45-100-F1
H20C	2,5	63	20	S1-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-100-F1
H21C	2,5	63	20	S1-31-32-33-34-35-36-37-38-39-40-41-42-43-44-45-46-47-48-49-100-F1
H35C	2,4	45	15	S1-60-32-54-42-40-50-34-36-37-38-44-52-46-45-100-F1
H45C	1,9	40	12	S1-53-34-35-36-37-38-58-47-46-45-49-100-F1
H55C	1,7	13	17	S1-39-35-50-33-61-31-55-54-43-44-52-46-64-45-63-49-100-F1
H65C	1,4	0	11	S1-54-43-44-56-58-62-47-52-48-49-100-F1
HDR	1,0	0	9	S1-55-58-62-44-48-49-63-45-100-F1
P	1,7	40	12	S1-36-38-34-40-42-43-59-58-56-57-48-100-F1

Obr. 9-1: Exportovaný soubor dat daných tratí z programu *Ocad*

9.1 Vstupní kontrola

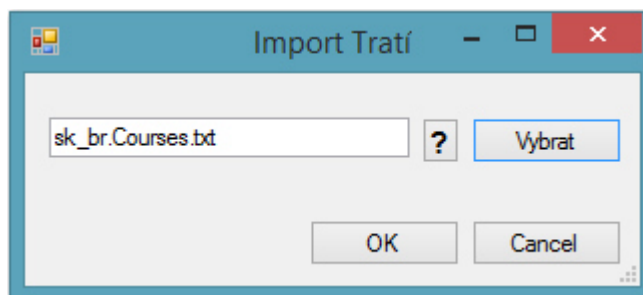
Každá posloupnost kontrol musí začínat vstupní kontrolou a tím bývá zpravidla startovní kontrola. Na obr. 9-1 lze vyzorovat, že *Ocad* generuje start pomocí symbolu „S“ s číselnou hodnotou, o jaký start se jedná. Z praktického hlediska se startovní kontrola používá pouze na začátku trati. Vstupní kontrolou může být i klasická kontrola, avšak z hlediska přehlednosti trati není vhodná.

9.2 Výstupní kontrola

Další speciálním případem je cílová kontrola. Tu lze identifikovat pomocí znaku „F“ v počátku názvu kontroly a z praktického hlediska je poslední kontrolou posloupnosti.

9.3 Načtení tratí ze souboru

Vhledem k velkému využívání programu *Ocad*, jsem se rozhodl pro načítání tratí do mé aplikace v jejich formátu exportování tratí. Uživateli pak v aplikaci stačí vybrat konkrétní soubor a data budou načtena do databáze.

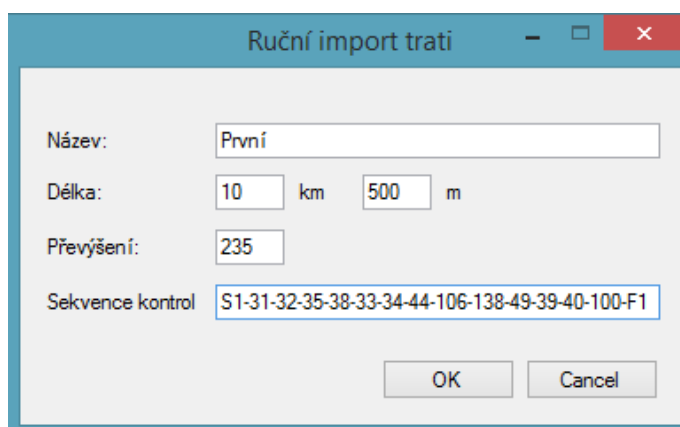


Obr. 9.3-1: dialogové okno pro import tratí ze souboru

Ke zjištění správného formátování dat je možno nahlédnout pomocí tlačítka „?“ . Pro úspěšné uložení je důležité vyplnit všechny atributy tratí. Délka tratí musí obsahovat klíčový znak „,“, oddělující kilometry a metry. Převýšení a počet kontrol musejí být číselné hodnoty. Jednotlivé kontroly v posloupnosti jsou odděleny znakem ‘-’. Jestliže v databázi existují požadované tratě, program vyzve ke smazání starých záznamů.

9.4 Ruční načtení tratí

Program nabízí i načtení tratí bez nutnosti externího souboru. Tento způsob je např. využitelný při trénincích, kdy jsou počty tratí malého rozsahu nebo není-li využit program *Ocad* pro tvorbu tratí.



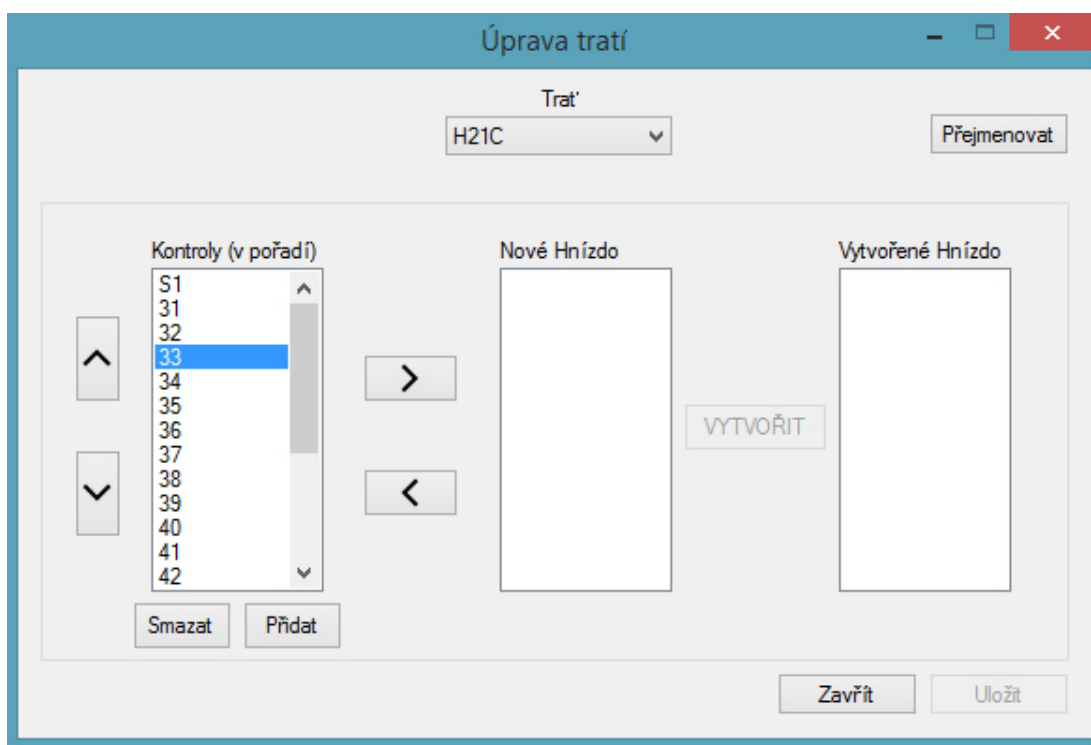
Obr. 9.4-1: dialogové okno pro ruční import tratí

Aplikace kontroluje, jestli trať se zvoleným názvem již v databázi neexistuje. Opět jsou kontrolovány číselné hodnoty v délce a převýšení trati. Pokud uživatel nevyplní v délce metry, automaticky bude doplněna hodnota 0. To samé platí i pro kolonku s převýšením.

Zadáváním posloupnosti kontrol, je pro oddělení kontrol potřeba použít symbol '-'. Dále je ověřena pozice startovní kontroly v posloupnosti – nesmí být použita jinde, než na prvním místě.

9.5 Úprava trati

Trať lze po jejím načtení upravovat. V dialogu je zobrazena příslušná trať a v prvním sloupci její posloupnost kontrol. Dílčí kontroly v uvedené posloupnosti lze mezi sebou jednoduše měnit pomocí šipek. Kontroly lze též přidávat a mazat pomocí daných tlačítek.



Obr. 9.5-1: dialogové okno pro úpravu trati

Je-li přidána kontrola, a navíc jde o kontrolu startovní, program jí automaticky přiřadí na začátek posloupnosti. Ostatní přidané kontroly se řadí automaticky na konec pole. Startovní kontrolu lze také přidat pouze v případě, jestliže není v dané posloupnosti již obsažena. Změna její pozice v posloupnosti je taktéž vyloučena.

Příklad: Funkce pro změnu indexu kontroly v posloupnosti kontrol trati

```
public void MoveItem(int direction)
{
    if (LB1.SelectedItem == null           ||           // kontrola není zvolena
        LB1.SelectedIndex < 0             ||           //
        LB1.SelectedItem.ToString()[0] == 'S' ||           // jde o startovní kontrolu
        LB1.SelectedItem.ToString()[0] == 's')

        return;

    // požadovaný index
    int newIndex = LB1.SelectedIndex + direction;

    if (newIndex < 0 || newIndex >= (LB1.Items.Count) || // nový index je mimo pole
        LB1.Items[newIndex].ToString()[0] == 'S'         || //nový index je na pol. st. kontroly
        LB1.Items[newIndex].ToString()[0] == 's' )

        return;

    //výměna hodnot mezi indexy
    string s = LB1.SelectedItem.ToString();

    LB1.Items[LB1.SelectedIndex] = LB1.Items[newIndex].ToString();
    LB1.Items[newIndex] = s;
    LB1.SelectedIndex = newIndex;
}
```

Dalším nástrojem v úpravě trati je tvorba „hnízd“, která slouží pro potřeby Scorelaufu (kapitola 2.4). Jde o skupinu kontrol, které budou moci být při závodě proběhnuty ve volném pořadí. Uživatel zadá požadované kontroly do druhého sloupce (určené pro nové hnízdo) a stiskne tlačítko pro vytvoření. Hnízdo se pak přidá na konec posloupnosti kontrol. Manipulovat s ním lze stejně jako s kontrolami. Vložené kontroly do hnízda lze najít kliknutím na něj. Pokud chce uživatel některou kontrolu z vytvořené skupiny vytáhnout, je potřeba přesunout položku hnízda opět do sloupce nové hnízdo. Poté lze požadovanou kontrolu vrátit zpátky do posloupnosti, nicméně zbylé kontroly se musí opět vytvořit jako skupina.

9.6 Smazání trati

Položka mazání tratí slouží k odstranění nepotřebných tratí. V dialogovém okně stačí vybrat název konkrétní trati a stisknout „OK“.

10 TVORBA STARTOVNÍCH ČASŮ

Nezbytnou potřebou tohoto typu softwaru je tvorba startovních časů pro závodníky. Pořadatel losuje startovní časy závodníků cca 2 dny před závodem, a ty pak umísťuje na webovou stránku závodu. Na níže položeném obrázku můžeme vidět základní parametry zvolené kategorie. Jedná se o čas startu, časový interval mezi závodníky, trať, typ startování a typ závodu. Je-li zvolen intervalový start, tak určujícím časem prvního losovaného závodníka je právě startovní čas kategorie. Každý další závodník je odstupňován zvoleným intervalem. V případě hromadného startu či startování pomocí SI kontroly mají všichni závodníci startovní čas roven času startu. Startuje-li se pomocí startovní krabičky, tak startovní čas slouží pouze jako informace, v jakém čase může první závodník odstartovat.

Obr. 10-1: Volitelné základní parametry kategorie

Typ závodu může být kvalifikace, rankingový nebo obecný závod. Zaměřme se nyní na volbu „kvalifikace“. V kapitole 8.10 bylo popsáno jak vytvořit kvalifikační skupiny. Nyní je potřeba specifikovat společnou kategorii. Do programu se napíše název nadřazené kategorie a zvolí se příslušné kvalifikační skupiny.

Obr. 10-2: Parametry pro kvalifikační závod

V případě změny, je potřeba stisknout tlačítko zrušit a provést požadované operace znovu. Ve startovní listině pak budou závodníci seřazení pod názvem nadřazené kategorie.

Společně s těmito základními vlastnostmi jsou v dialogu důležité informace kategorie, sloužící uživateli, vytvářející startovní listinu. Každou změnou vlastnosti se informace mění. Kategorie, které startují ve stejné minutě, by podle pravidel neměly mít stejnou první kontrolu. Z těchto údajů lze tuto informaci zjistit. Kategorie, které mají společnou trať, by neměly startovat ve stejné minutě. Z informací lze vyčíst interval pro odstartování celé kategorie se společnou tratí. Dochází-li při losování k této kolizi, program nabídne posunutí startovní času losované kategorie za kategorií s ní kolidující.

The dialog box 'Informace' contains the following fields:

- První kontrola:** Input field with the value '58'.
- Společný start s kat.:** A list box containing the following categories: D10, D12, D14, D21B, D21C, D35, D65.
- Náběh na stejnou 1. kontrolou ve stejné startovní minutě s kat.:** A list box containing D21B and D21C.
- Kategorie se společnou tratí a s jejím časovým intervalem:** A list box containing:
 - D21B -> 20:27 - 22:15
 - D21C -> 20:27 - 22:21

Obr. 10-3: Důležité informace pro tvorbu startovek

Losování kategorií je možné pouze tehdy, mají-li přiřazenou konkrétní trať. V dialogovém okně lze pozorovat jejich statusy. Nemá-li kategorie přiřazenou trať, svítí u ní kontrolka červenou barvou. Pokud je trať přiřazena a kategorie prozatím není losována, kontrolka je žluté barvy. Úspěšné losování je pak značeno barvou zelenou.

Kategorie	Status
D10	●
D12	●
D14	●
D21A	●
D21B	●
D21C	●
H45	●

Obr. 10-4: Statusy vytvořených kategorií

Příklad: Nahrání informace – kategorie se společnou tratí a jejich interval pro odstartování

```
kat.read_data(cn, "etapaID", id.etID );

for (int i = 0; i < kat.kat_data.Count; i++)
{
    if (kat.kat_data[i].kategorie != c_kategorie.SelectedItem.ToString() &&
        kat.kat_data[i].trat == c_trat.SelectedItem.ToString() &&
        kat.kat_data[i].typ_st == "Intervalový")
    {
        int poc_st = Int32.Parse(kat.kat_data[i].pocet_st)
        int st_int = Int32.Parse(kat.kat_data[i].start_int); int interval = poc_st * st_int;
        int hod = interval / 60;
        int min = interval % 60;
        String s_hod = "";
        String s_min = "";
        int cilovy_hod = Int32.Parse(kat.kat_data[i].start.Substring(0, 2)) + hod;
        int cilovy_min = Int32.Parse(kat.kat_data[i].start.Substring(3, 2)) + min;

        if (cilovy_min > 59)
        {
            cilovy_min -= 60;
            cilovy_hod++;
        }

        if (cilovy_hod > 23)
            cilovy_hod -= 24;

        if (cilovy_hod < 10)
            s_hod = "0" + cilovy_hod.ToString();
        else
            s_hod = cilovy_hod.ToString();

        if (cilovy_min < 10)
            s_min = "0" + cilovy_min.ToString();
        else
            s_min = cilovy_min.ToString();

        LB_spolTr.Items.Add(kat.kat_data[i].kategorie + " -> " + kat.kat_data[i].start +
            " - " + s_hod + ":" + s_min);
    }
}
```

Program načte z databáze všechny vytvořené kategorie v dané etapě. Poté všechny načtené prochází v cyklu. Žádoucí jsou pouze kategorie se společnou tratí a v rámci intervalového startu. Taktéž se nesmí jednat o právě zvolenou kategorii, se kterou se pracuje. Pokud je podmínka splněna, je načten počet startujících a startovní interval kategorie. Ty se pak mezi sebou násobí a vznikne celkový čas pro odstartování celé kategorie. Tento čas je pak přičten ke startovnímu času prvního závodníka v kategorii a je získán čas posledního startujícího. Je nutno ošetřit, překročí-li čas povolenou mez, resp. 60 minut a 24 hodin. Je-li počet minut a hodin menší než číslo 10, je přidána přípona „0“.

10.1 Losování závodníků

V okně pro losování se ladí detaily okolo účastníků. V pravidlech je sice nařízeno losování z náhodného generátoru, avšak některé požadavky účastníků bývají akceptovány. Typickým požadavkem je rozdělení startu „manželského páru“, kdy jde o hlídání malého dítěte na shromaždišti závodu. Jeden z nich pak bývá umístěn na začátku startovní listiny, druhý na jejím konci. Požadavky jsou součástí přihlášek daných osob. Kdyby tyto případy požadavků měly povinný formát, šlo by vytvořit i automatické rozdělení, bohužel každá poznámka je unikátem. V dané tabulce je možné tyto informace vyčíst. Je tedy možné ručně příslušné závodníky vložit do speciální skupiny, pomocí kterých se pak budou automaticky řadit na začátek nebo konec startovní listiny.

Pokud jsou vytvořeni vakanti, je pro ně třeba specifikovat typ řazení se do startovní listiny. Řadit vakanty lze rovnoměrně do startovky, tzn. třetina z nich na začátek, další na střed a poslední na konec startovky. Dalšími možnostmi jsou řazení na začátek, střed a konec startovky.

The screenshot displays a web interface for managing race start lists. It is divided into several sections:

- Závodníci (Racers):** A list of racers with their IDs and names. A 'Smazat' (Delete) button is at the bottom.
- Seznam závodníků bez požadavku:** A list of racers without specific requirements.
- Seznam závodníků pro brzký start:** A list of racers for an early start, currently containing CTB8151 Kamila Gregorová and VSP7850 Barbora Baldrianová.
- Seznam závodníků pro pozdní start:** A list of racers for a late start, currently containing CHT8353 Kateřina Matrasová.
- Vakanti (Vacancies):** A section for managing vacancies, including a dropdown for sorting order (currently 'Dopředu') and a 'Přidat' (Add) button.
- Poznámky závodníků (Racer Notes):** A list of notes for specific racers, such as 'CTB8151 Kamila Gregorová - rozdílný start s Gregor Mar'.
- Startovní listina (Start List):** A vertical list of racers with their assigned start times, ranging from 20:27 to 21:13. The list includes 'Vakant' entries at the beginning.

Navigation buttons 'LOSOVAT' (Generate) and 'ZRUŠIT' (Cancel) are located at the top right. Arrows on the left side of the start list allow for manual reordering.

Obr. 10.1-1: Okno pro losování

Jestliže jsou data připravena k losování, postačí stisknout tlačítko losovat. V pravé části okna se vygenerují startovní časy pro jednotlivé závodníky. Pomocí šipek lze ručně měnit startovní pozice závodníků resp. jejich časy. Chce-li uživatel celé losování změnit, musí použít tlačítko zrušit, které losovaná data vymaže.

Příklad: Nahrání vakantů do příslušných míst startovky typu – dopředu, na střed nebo dozadu

```
// Jiné řazení, než rovnoměrný typ
if (c_raz_vak.SelectedIndex <= 2)
{
    // nahrání startovního indexu, kde začít vkládat vakanty
    switch (c_raz_vak.SelectedIndex)
    {
        case 0: index = 0;
                break;
        case 1: index = LB_startList.Items.Count / 2;
                break;
        case 2: index = LB_startList.Items.Count;
                break;
    }
    //Cyklus prochází tabulku se závodníky, jakmile narazí na vakanta, přidá jej do startovní
    //listiny
    for (int i = 0; i < LB_zavodnici.Items.Count; i++)
    {
        if (LB_zavodnici.Items[i].ToString().Substring(0, 1) == " ")
        {
            LB_startList.Items.Insert(index, "".PadLeft(max + 1) + "Vakant");
            index++;
        }
    }
}
```

Pravidla též hovoří o tom, že závodníci z jednoho klubu by neměli startovat za sebou, je-li to z matematického hlediska možné. V průběhu losování program tento požadavek ošetří.

Příklad: Nahrání závodníků do startovky pomocí náhodného generátoru s ošetřením, aby dva závodníci stejného klubu nezávodili za sebou

```
Random r = new Random();
// procházení seznamu závodníků
while (l_zavodnici.Count > 0)
{
    // generování pozice
    int i = r.Next(0, l_zavodnici.Count);

    // ošetření zda-li nejsou losování 2 závodníci ze sebou ze stejného oddílu
    if (kluby.Count == 0 ||
        (kluby[kluby.Count - 1] != l_zavodnici[i].klub ||
         l_zavodnici.GroupBy(x => x.klub).Distinct().Count() == 1)
    )
    {
        // přidání klubu posledního losovaného závodníka do seznamu
        kluby.Add(l_zavodnici[i].klub);
        LB_startList.Items.Add(l_zavodnici[i].reg_cislo + " " +
                               l_zavodnici[i].jmeno + " " +
                               l_zavodnici[i].prijmeni );
        l_zavodnici.RemoveAt(i);
    }
}
```


11 SBĚR DAT Z ČIPŮ

11.1 Systém SPORTident

V kapitole 2.2 byl ve zkratce popsán tento systém. K tomu bych doplnil, že k úspěšnému zvládnutí závodu (z hlediska pořadatele) je potřeba využít celé sady. Do sady patří mimo kontrol pro stanoviště také kontroly pro smazání a ověření úspěšného smazání záznamu nebo startovací kontroly. Pro načtení záznamu z čipu je využito speciální kontroly s datovým kabelem typu USB nebo RS-232.



Obr. 11.1-1: kontrola pro vyčítání čipů s USB kabelem [9]

11.2 Závodní průkazy

Závodním průkazem resp. čipem musí být vybaven každý závodník, chce-li být klasifikován. V dnešní době existuje již mnoho řad, které s trochou nadsázky přidělávají vrásky na čele vývojářům těchto typů softwaru.



Obr. 11.2-1: Závodní průkaz při průchodu kontrolou [10]

Specifikace jednotlivých sérií je možné vidět v příloze. Každá série čipů má jinou kapacitu pro uložení dat a jinou rychlost načítání dat. Nejstarší série 5 a 6 ukládají čas ve formátu 12 hodin. Ostatní série už běží na formátu 24 hodin. Starší verze taktéž neukládají údaje (čas a kód kontroly) při mazání starých dat a jejich ověření. Aplikace by nyní měla umět komunikovat se všemi vydanými sériemi.

11.3 Komunikace se systémem SPORTident

V této aplikaci je důležité umět komunikovat s krabičkou pro sběr dat z čipu. Firma SPORTident nabízí vývojářům, ke komunikaci se systémem, vlastní soubor s knihovnou pod názvem *SICOMM*. Soubor „dll“ knihovny je potřeba vložit do složky s vlastním softwarem a poté načíst do projektu ve vývojovém prostředí.

Příklad: Připojení se ke stanici pro sběr dat

```
public SiComm siComm = new SiComm();
public bool chyba = false;

siComm.Error +=new __SiComm_ErrorEventHandler(siComm_Error);

//funkce, která je spuštěna při neúspěšném připojení
void siComm_Error(ref string No, ref string Description)
{
    chyba = true;
}

siComm.Initialize(Int32.Parse(TB.Text), 38400, 0, "");

if( chyba == false)
{
    siComm.PortOpen = true;
    siComm.SetModeDirect();
    siComm.BeepIfStationReady(2);
    siComm.SetSiS_ExtendedProtocol_NewBoard(Set_Type.set_yes);

    MessageBox.Show("SI Vyčítací jednotka byla připojena", "Upozornění",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
}
else
{
    chyba = false;
    MessageBox.Show("Nepodařilo se připojit ke zvolenému portu", "Upozornění",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    return;
}
```

V metodě pro inicializaci je důležité zvolit správné číslo portu, kde máme připojenou stanici a hodnotu parity. Program se pokusí ke stanici připojit. Případný neúspěch je zachycen událostí a poté je uživateli vypsána zpráva o nezdařilém připojení.

Podaří-li se připojit, stanice vydá 2krát zvukový a světelný signál a následně bude svítit permanentně. Je-li ukončeno spojení, stanice zhasne. Program uživateli nabízí připojení až třech krabiček pro vyčtení. Možností je i automatické načtení krabičky bez nutnosti informace, na jakém portu se nachází.

Příklad: automatické připojení ke stanici pro sběr dat

```
string[] ports = SerialPort.GetPortNames();

for (int i = 0; i < ports.Count(); i++)
{
    VyberPortu vp = new VyberPortu();

    if (siComm1.PortOpen == false)
    {
        siComm1 = vp.open_port(Int32.Parse(ports[i].Substring(3)));

        if (siComm1 != null)
        {
            siComm1.ReceiveComplete +=
                new __SiComm_ReceiveCompleteEventHandler(siComm1_ReceiveComplete);
            break;
        }
    }
}
```

Program zjistí všechny možné sériové porty. Následně se pokusí k některému z nich připojit. Při úspěšném připojení přidá událost pro přijímání dat a cyklus ukončí.

11.4 Zjištění konkrétní série čipu

Jakmile je krabička připojena, je vše připravené pro nahrání dat ze závodního průkazu. Jak už bylo zmíněno, na trhu je k dispozici široká škála různých sérií čipů. Vložením čipu do krabičky se vyvolá událost, do které nejdříve „dorazí“ kód o dané sérii. Kód je v hexadecimálním tvaru. Po nalezení série se událost znovu vyvolá, nyní už pro uložení dat z čipu.

Příklad: zjištění čipu řady 8,9, pCard nebo tCard

```
switch (cmd)
{
    //vlozeni cipu rady 8,9 pCard, tCard
    case 0xE8:
        bReadyToRead1 = true;
        siComm1.Clear_CardData();
        siComm1.Read_SiC_CardX(SI_Comm.CardXCardRead_Type.CardX_Punch_BlocksP);
        break;
}
```

11.5 Načtení dat z čipu

Dialogové okno pro vyčítání je rozloženo na dvě části. V první části jsou vypisovány informace o načítaném čipu a ve druhé je vypsán seznam chyb způsobený určitou poruchou při načítání. Ještě před samotným načtením potřebných údajů z čipu, program zjistí, zdali je číslo vloženého čipu evidováno v databázi. V případě neúspěšného nalezení je zhlášena chyba. Číslo neznámého čipu je však vypsáno a lze jej dodatečně přiřadit konkrétnímu závodníkovi.

Vyčítací jednotka		Tisk							
Reg. číslo	Jméno	Příjmení	Číslo čipu	Výsledný čas	Upravit	Smazat	Vrátit čip	Status	
VL18901	Tomáš	Kobr	1969696	51:23	upravit	smazat	ne	OK	
Seznam chyb									

Obr. 11.5-1: Dialogové okno pro vyčítání závodních průkazů

Dalšími ověřeními jsou přiřazení startovního času závodníka, načtení trati nebo opětovné načtení dat z čipu. Není-li nalezen žádný problém, program přečte potřebná data z průkazu (časy a kódy kontrol).

Data ze speciálních stanic (mazací, ověřovací, startovní a cílová krabička) jsou uložena do databáze samostatně oproti kontrolám z trati, které jsou načtené do společného seznamu. Aplikace načte posloupnost kontrol trati spojenou s číslem čipu a postupně jí prochází, viz obr. 11.5-2. První kontrolu trati hledá v seznamu zjištěných kontrol v čipu spolu se startovní a cílovou. Pokud bude nalezena, tak zapíše index, na kterém se kontrola nachází. Stejným způsobem jsou nalezeny další kontroly s tím, že se začíná poté hledat od indexu předchozí nalezené kontroly. Jestliže není kontrola nalezena, zapíše se na index kód -999. Startovní a cílová kontrola jsou vedeny mimo

seznam, a proto se k identifikaci místo indexu použije číslo -1 pro startovní a -2 pro cílovou kontrolu.

Seznam kódů kontrol zaznamenané v čipu mimo startovní a cílové kontroly

31 , 36 , 32

index : 0 , 1 , 2

Startovní kontrola - S1 ; index : -1

Cílová kontrola - F1 ; index : -2

Posloupnost kontrol zaznamenaná v čipu	S1 , 31 , 36 , 32 , F1
Posloupnost kontrol příslušné trati	S1 , 31 , 35 , 36 , 32 , F1
Přiřazené indexy	-1 , 0 , -999 , 1 , 2 , -2

Obr. 11.5-2: Příklad uložení indexů zjištěných kontrol z čipu

Z výše uvedeného příkladu je patrné, že v závodním průkazu není nalezena kontrola č. 35, proto bude závodník s tímto číslem čipu diskvalifikován.

Další fází je nahrání mezičasů mezi nalezenými kontrolami. Ze znalosti indexů, porovnáme vždy časy dvou sousedních kontrol a odečteme mezi sebou. Pokud není čas z jedné kontroly dané dvojice k dispozici, není mezičas načten. Jestliže je rozdíl obou časů záporný, opět se mezičas nezapíše. K tomu by mohlo např. dojít v případně špatného seřazení času ve všech kontrolách. Výjimkou je skok přes 24 hod, který je v programu ošetřený.

Příklad: načtení mezičasu mezi kontrolami

```

public string mezicas(string cas_s, string cas_f)
{
    //ošetření, jestli jeden z časů nechybí
    if (cas_s == "" || cas_f == "")
        return "";

    //převod na sekundy
    int cf_sek = Int32.Parse(cas_f.Substring(0, 2)) * 3600 +
                Int32.Parse(cas_f.Substring(3, 2)) * 60 +
                Int32.Parse(cas_f.Substring(6, 2));

    int cs_sek = Int32.Parse(cas_s.Substring(0, 2)) * 3600 +
                Int32.Parse(cas_s.Substring(3, 2)) * 60 +
                Int32.Parse(cas_s.Substring(6, 2));

    String mezicas = "";
    int rozdil;

    if (cf_sek > cs_sek)
        rozdil = cf_sek - cs_sek;
    else
        rozdil = cf_sek + (86400 - cs_sek);

    //maximální povolený rozdíl časů je 999min
    if (rozdil > 59940)
        return "";

    //minuty
    if (((rozdil / 60) < 10))
        mezicas += "0" + (rozdil / 60).ToString() + ":";
    else
        mezicas += (rozdil / 60).ToString() + ":";
    //sekundy
    if (((rozdil % 60) < 10))
        mezicas += "0" + (rozdil % 60).ToString();
    else
        mezicas += (rozdil % 60).ToString();
    return (mezicas);
}

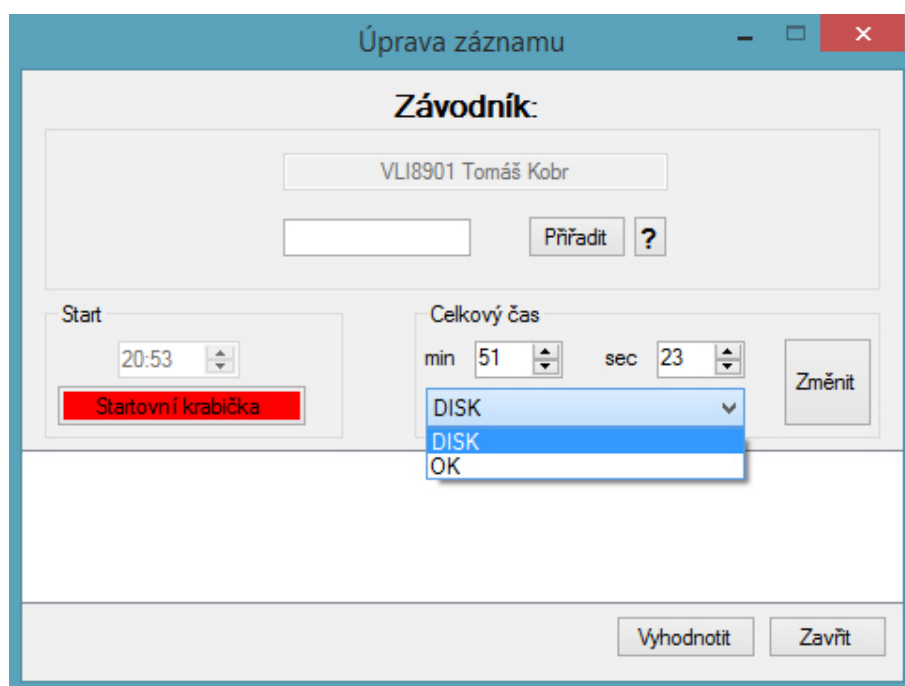
```

11.6 Úprava načteného záznamu

Zjištěný záznam je možné do jisté míry upravit. Může se stát např., že některá kontrola na trati vypadne a závodník je nucen označit kontrolu pomocí mechanických kleští. Tím pádem bude záznam v čipu chybět a program vyhodnotí závodníka jako diskvalifikovaného. Pořadatel má možnost v úpravě záznamu takové závodníka ručně zrušit diskvalifikaci. Naopak, kdyby se závodník na trati zachoval některým nepovoleným způsobem, je možné dodatečně osobu diskvalifikovat. Měnit lze i výsledný čas běže.

V některém z existujících programů bylo vidět, že program nerozeznal jednu důležitou věc. Závodníkovi se automaticky přepsal přidělený startovní čas, jestliže označil startovní krabíčku. Hodnotu času vzal právě z kýžené startovní krabíčky. Tato aplikace má tento problém vyřešený. Může se však stát, že závodník z nějakého důvodu nestíhá odstartovat v jeho přiděleném startovním čase. Pořadatel mu posléze umožní odstartovat na startovní krabíčku. Program sice nebude reagovat na jeho změnu startovacího typu, avšak v tom dialogu je možné změnit jeho typ startu, a tím bude změněn i jeho startovní čas.

V poslední řadě je příležitost přiřadit neznámý čip konkrétní osobě. K tomu postačí zadat registrační číslo závodníka.



Obr. 11.6-1: dialogové okno pro úpravu vyhodnoceného záznamu

11.7 Tisk mezičasů

Po načtení dat z čipu je závodníkovi vytisknut „lístek“ s průběhem jeho působení na trati. V hlavičce je vypsán název závodu a konkrétní údaje o závodníkovi. V další části je vypsán výsledný čas. Následuje část s mezičasy dílčích kontrol v dané posloupnosti. Jeden ze sloupců ukazuje absolutní časy od startovního času, pro dané stanoviště. K nim jsou přiřazeny ztráty na vítězného závodníka a taktéž příslušná pozice. V dalším sloupci jsou zobrazeny jednotlivé mezičasy mezi kontrolami. Ztráty a pozice jsou vypsány obdobně jako v předchozím sloupci.

```

-----OCAD.cz cup-----
Jan SCHULHOF
OOB TJ TURNOV                                TUR8900
H21L                                           1969696
S:15:17:01 CH:15:16:12 F:15:31:59
-----
                                14:58
-----
1  70  00:27 + 00:03  3  00:27 + 00:03  3
2  71  01:15 + 00:02  2  00:48 + 00:00  1
3  72  02:17 + 00:00  1  01:02 + 00:02  3
4  73  02:46 + 00:00  1  00:29 + 00:01  2
5  74  03:12 + 00:00  1  00:26 + 00:00  1
6  75  03:37 + 00:00  1  00:25 + 00:00  1
7  72  03:53 + 00:00  1  00:16 + 00:00  1
8  77  04:43 + 00:00  1  00:50 + 00:00  1
9  78  05:40 + 00:00  1  00:57 + 00:00  1
10 79  07:04 + 00:00  1  01:24 + 00:00  1
11 81  07:45 + 00:00  1  00:41 + 00:05  6
12 80  09:38 + 00:00  1  01:53 + 00:00  1
13 82  10:47 + 00:00  1  01:09 + 00:00  1
14 84  11:15 + 00:00  1  00:28 + 00:20  20
15 87  13:04 + 00:24  5  01:49 + 00:43  20
16 88  13:36 + 00:23  5  00:32 + 00:00  1
17 83  14:01 + 00:29  5  00:25 + 00:07  11
18 100 14:48 + 00:26  5  00:47 + 00:05  2
Finish: 14:58 + 00:27  5  00:10 + 00:04  6
-----
Ztráta na nejlepší mezičasy:      + 01:30
Pozice v cíli: 5/22

Průběžné výsledky v kategorii:
1  Wolf      Daniel      14:31 + 00:00
2  Luštický  Martin     14:33 + 00:02
3  Nývlt     Martin     14:38 + 00:07
-----ocad team-----

*****kobrIS*****
Tomáš Kobr

```

Obr. 11.7-1: „lístek“ s průběhem závodu konkrétní osoby

Další prostor je věnován statistikám, jako celkový čas ztrát na ideální mezičasy mezi kontrolami a průběžná pozice v kategorii z dosud doběhnutých závodníků. Pro informaci jsou vypsáni tři nejlepší osoby s jejich výslednými časy a rozestupy, samozřejmě pokud existují. V závěru je vypsán pořadatelský tým a název softwaru. Nastavení tisku bude vysvětleno v další kapitole.

Příklad: seřazení mezičasů od nejlepšího k nejhoršímu na konkrétní kontrole

```
List<int> mez = new List<int>();
//přiřazení mezičasu od všech závodníků z dané kontroly
for (int j = 0; j < l_data.Count; j++)
{
    if (rel_mez == true) //relativní mezičasy
        mez.Add(l_data[j].l_mezicasy[i - 1].c_cas);

    else //absolutní mezičasy
    {
        //ověření diskvalifikace
        if (i != 1 && l_data[j].l_mezicasy[i - 2].pozice2 == "-1")
        {
            mez.Add(90000);
            l_data[j].l_mezicasy[i - 1].c_cas2 = 90000;
        }

        //na pozici kontroly, kdy už je diskvalifikace
        else if (l_data[j].l_mezicasy[i - 1].celkovy_cas == "")
            mez.Add(90000);
        else
            mez.Add(l_data[j].l_mezicasy[i - 1].c_cas2);
    }
}
mez.Sort(); //třídění mezičasů od nejlepšího
```

Program postupně prochází všechny závodníky a ukládá daný mezičas. Pokud mezičas chybí, nahraje do něj hodnotu specifickou hodnotu 90000, kterou nemůže žádný mezičas dosáhnout. Čas je již předtím převeden na číselnou hodnotu pro pozdější snadné třídění. Je též zapotřebí specifikovat, o jaký typ mezičasu se jedná, z důvodů odlišných testů.

Příklad: nahrání pozice závodníka na dané kontrole

```
for (int j = 0; j < l_data.Count; j++)
{
    int pozice;
    //nahrání příslušné pozice v seznamu mezičasů, dle znalosti mezičasu závodníka
    if (rel_mez == true)
        pozice = mez.IndexOf(l_data[j].l_mezicasy[i - 1].c_cas);
    else
        pozice = mez.IndexOf(l_data[j].l_mezicasy[i - 1].c_cas2);
    //posun pozice, pokud závodník předním disponuje stejným mezičasem
    for (; pozice >= 0; pozice--)
        if (pozice == 0 || mez[pozice] == 90000 || mez[pozice - 1] != mez[pozice])
            break;

    if (mez[pozice] == 90000) //chybějící mezičas
    {
        if (rel_mez == true)
            l_data[j].l_mezicasy[i - 1].pozice = (-1).ToString();
        else
            l_data[j].l_mezicasy[i - 1].pozice2 = (-1).ToString();
    }
}
```

```

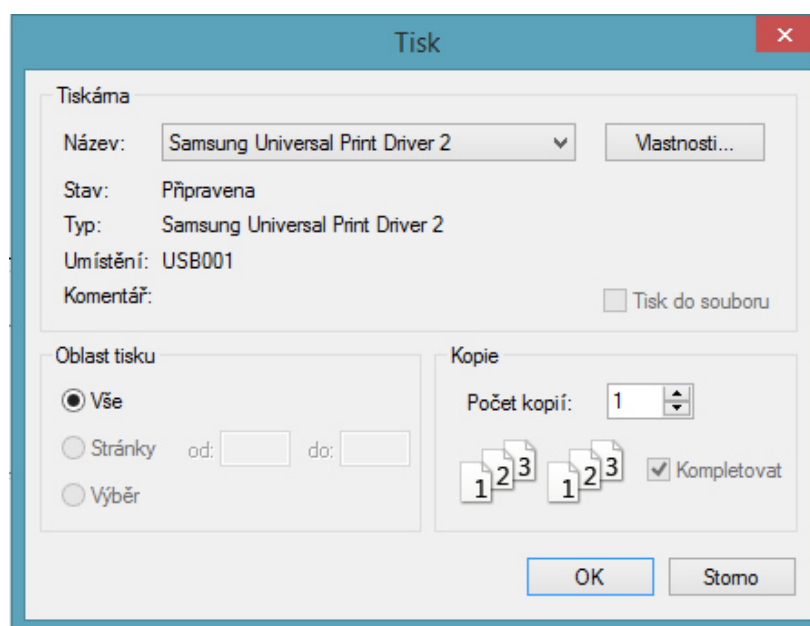
else
{
  if (rel_mez == true)
    l_data[j].l_mezicasy[i - 1].pozice = (pozice + 1).ToString();
  else
    l_data[j].l_mezicasy[i - 1].pozice2 = (pozice + 1).ToString();
}

```

Ze tříděného seznamu mezičasů na konkrétní kontrole dochází přiřazení pozice příslušnému závodníkovi. Pokud mají někteří závodníci stejný čas, musí dojít ke korekci dané pozice. Pokud není mezičas nalezen (hodnota 90000) přiřadí se specifická hodnota pozice „-1“.

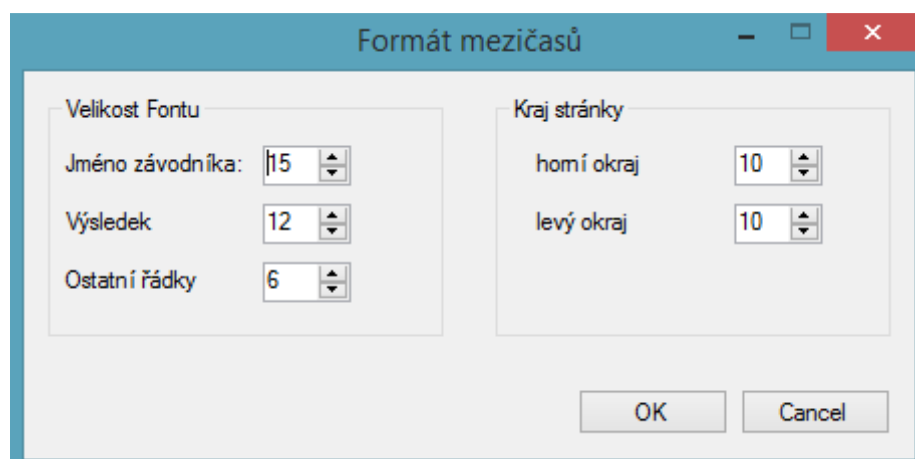
11.8 Nastavení tisku

Před samotným tiskem je nutno nastavit jeho parametry. Pro práci s tiskem nabízí Visual studio speciální komponentu. V dialogovém okně probíhá nastavení obdobně, jako např. nastavení tisku ve wordu.



Obr. 11.8-1: dialogové okno pro nastavení tisku

Důležité je hlavně zvolit správnou tiskárnu a formát papíru. Pořadatelské týmy v dnešní době hodně využívají pokladních tiskáren, čemuž je i přizpůsobený formát výstupního „lístku“. Je-li např. využit tisk na formátu A4 je možnost zvolit více závodníků na papír, kvůli jeho úspoře. Jednotlivé záznamy se pak rozdělí např. pomocí nůžek. Uživatel má též možnost měnit velikost znaků na záznamu a odsazení z pravého horního rohu papíru.



Obr. 11.8-2: dialogové okno pro nastavení formátu záznamu

Vykreslení vytvořených dat je zprostředkováno prostřednictvím třídy *CANVAS*. Tato třída je specifická pro kreslení. Na dané souřadnice jsou pak vykresleny potřebné údaje.

12 VÝSLEDKY

V sekci s výsledky lze nahlédnout na všechny závodníky, kteří si vyčetli svůj závodní průkaz. V horní liště lze jednoduše přepínat mezi vytvořenými kategoriemi. K dispozici je náhled celkových výsledků a mezičasů. Pro obnovení dat z databáze lze využít příslušného tlačítka. Záznamy je možné do jisté míry upravovat stejně tak, jak bylo popsáno v kapitole 11.6. V nabídce je také možnost danou osobu ručně odstranit.

Výsledky [X]

Celkové Výsledky H21L

Pořadí	Příjmení	Jméno	Klub	Reg. číslo	Licence	Číslo čipu	Výsledný čas	Ztráta	Upravit	Smazat
1	Wolf	Daniel	OOB TJ TURNOV	TUR9100	R	515499	14:31	00:00	Upravit	Smazat
2	Luštický	Martin	OOB TJ TURNOV	TUR8700	R	2035709	14:33	00:02	Upravit	Smazat
3	Nývlt	Martin	ODDÍL OB KOTLÁŘKA	DKP8200	C	1207407	14:38	00:07	Upravit	Smazat
4	Pospíšil	Ondřej	ODDÍL OB KOTLÁŘKA	DKP8717	C	7202874	14:51	00:20	Upravit	Smazat
5	Schulhof	Jan	OOB TJ TURNOV	TUR8900	C	1969696	14:58	00:27	Upravit	Smazat
6	Váňa	Martin	OOB TJ TURNOV	TUR8401	C	1840303	15:12	00:41	Upravit	Smazat
7	Kolín	Jan	ODDÍL OB KOTLÁŘKA	DKP7701	C	9007701	15:42	01:11	Upravit	Smazat
8	Hošek	Radim	KOB DOBRUŠKA	DOB8737	C	301627	15:55	01:24	Upravit	Smazat
9	Martan	Jiří	SLAVIA LIBEREC ORIENTEERING	VL17300	C	8129	16:30	01:59	Upravit	Smazat
10	Burda	Matěj	OK JISKRA NOVÝ BOR	BOR9608	A	7667667	16:49	02:18	Upravit	Smazat
11	Kolaja	Jan	SLAVIA LIBEREC ORIENTEERING	VL18100	C	2067313	17:05	02:34	Upravit	Smazat

Obr. 12-1: náhled na celkové výsledky

Výsledky [X]

Mezičasy H21L

Poř.	Závodník	Čas	70	71	72	73	74	75
1	TUR9100 Daniel Wolf OOB TJ TURNOV	14:31	00:26 (2)	00:50 (3)	01:02 (3)	00:29 (2)	00:26 (1)	00:25 (1)
2	TUR8700 Martin Luštický OOB TJ TURNOV	14:33	00:30 (8)	00:57 (7)	01:01 (2)	00:28 (1)	00:26 (1)	00:29 (8)
3	DKP8200 Martin Nývlt ODDÍL OB KOTLÁŘKA	14:38	00:24 (1)	00:49 (2)	01:06 (6)	00:29 (2)	00:26 (1)	00:28 (5)
4	DKP8717 Ondřej Pospíšil ODDÍL OB KOTLÁŘKA	14:51	00:27 (3)	00:51 (4)	01:00 (1)	00:31 (7)	00:31 (9)	00:28 (5)
5	TUR8900 Jan Schulhof OOB TJ TURNOV	14:58	00:27 (3)	00:48 (1)	01:02 (3)	00:29 (2)	00:26 (1)	00:25 (1)

Obr. 12-2: náhled na mezičasy

V okně pro mezičasy jsou vidět hodnoty časů na jednotlivých kontrolách a k nim příslušné pozice. V prvním řádku se jedná o mezičasy mezi kontrolami, v dalším jde o mezičasy počítané od startu závodu. Zvýrazněna je vedoucí pozice na daném mezičase.

13 EXPORTY DAT

Aplikace nabízí širokou nabídku exportu vytvořených dat. Exportovat lze informace o přihlášených osobách nebo startovní listiny, oboje je možné zvolit v uspořádání po kategoriích nebo klubech. Dalšími daty k exportování jsou údaje o platbách, výsledky a mezičasy. Exportovaná data je možné uložit ve formátu PDF nebo HTML s výjimkou, že platby lze exportovat pouze v PDF formátu. Speciálním exportem jsou výsledky pro ORIS ve formátu XML.

13.1 Export do PDF

K tvorbě PDF souboru je využito knihovny *iTextSharp*. Pomocí příslušných objektů je pak soubor vytvořen. Soubory mají na každé straně vytvořeno hlavičku s údaji o typu exportovaných dat, názvu závodu, názvu programu, stránce a časem vytvoření. Další část představují informace exportovaných dat.

Konečné výsledky	OCAD.cz cup kobriS © Tomáš Kobr	Stránka 9 3. 3. 2015 11:49:42
------------------	------------------------------------	----------------------------------

H21L (28/28) - 2,4km - 80m - 18K

1	Wolf	Daniel	OOB TJ TURNOV	TUR9100	R	14:31	+	00:00
2	Luštický	Martin	OOB TJ TURNOV	TUR8700	R	14:33	+	00:02
3	Nývlt	Martin	ODDÍL OB KOTLÁŘKA	DKP8200	C	14:38	+	00:07
4	Pospíšil	Ondřej	ODDÍL OB KOTLÁŘKA	DKP8717	C	14:51	+	00:20
5	Schulhof	Jan	OOB TJ TURNOV	TUR8900	C	14:58	+	00:27
6	Váňa	Martin	OOB TJ TURNOV	TUR8401	C	15:12	+	00:41
7	Kolín	Jan	ODDÍL OB KOTLÁŘKA	DKP7701	C	15:42	+	01:11

Obr. 13.1-1 náhled exportovaných dat v PDF souboru

Příklad: tvorba PDF souboru

```
//vytvoření instance pro zápis do PDF
PdfWriter pdfWriter;
//vytvoření dokumentu
Document doc = new Document(PageSize.A4, 40f, 10f, 105f, 30f);
//vytvoření datového proudu
msReport = new FileStream(path, FileMode.Create);
//Vytvoření fontu
bf = BaseFont.CreateFont(ARIALUNI_TFF1, BaseFont.IDENTITY_H, BaseFont.NOT_EMBEDDED);
f = new iTextSharp.text.Font(bf2, 8, iTextSharp.text.Font.NORMAL); pdfWriter = PdfWriter.GetInstance(doc, msReport);
```

```

using (doc)
{
    doc.Open(); //otevření proudu
    doc.Add(new Phrase("Ahoj", f)); //vypsání textu
    doc.Add(new Phrase("\n", f)); //odřádkování
    doc.Close(); //uzavření proudu
}
Process.Start(path); //otevření souboru

```

13.2 Export do HTML

K vytvoření tohoto typu souboru je využito klasického zápisu textu do souboru. Aby byl soubor použitelný na internetu, je potřeba využít syntaxe jazyka HTML.

Příklad: zápis textu pro soubor HTML

```

using (StreamWriter sw = new StreamWriter(path, false, Encoding.Default))
{
    string jmeno = "Tomáš Kobr";
    sw.WriteLine("<font face=\\\"Consolas\\\"> " + jmeno + "</font>");
}

```

13.3 Export výsledků pro ORIS

ORIS nabízí nahrání výsledků pomocí externího souboru v různých strukturách dat. Moje aplikace využije vytvoření dat ve formátu XML. Tvorba souboru probíhá obdobným stylem, jako v předchozím případě. Vytvořený soubor pak pořadatel vloží ručně do ORISU.

```

1 <?xml version="1.0" encoding="windows-1250"?>
2 <!DOCTYPE ResultList SYSTEM "IOFdata.dtd">
3 <ResultList>
4     <IOFVersion version="2.0.3"/>
5     <ClassResult>
6         <ClassShortName>D10</ClassShortName>
7         <PersonResult>
8             <Person>
9                 <PersonName>
10                    <Family>Matysov</Family>
11                    <Given>Kristina</Given>
12                </PersonName>
13                <PersonId>0555</PersonId>
14                <CountryId>CHA</CountryId>
15            </Person>

```

Obr. 13.3-1 náhled vytvořených dat formátu XML

14 TESTOVÁNÍ APLIKACE

První ostrý test se uskutečnil přímo v areálu Technické Univerzity v Liberci, prostřednictvím veřejného závodu v orientačním běhu s názvem OCAD cup. Závod proběhl dne 1.3.2015 se startem ve 14:30. Jednalo se o sprint se způsobem startování na startovní krabičku, tudíž mohl závodník vystartovat v libovolném čase.

Závodníci byli přihlášení přes systém ORIS, proto nahrání osob do databáze proběhlo přes webový server. Celkem bylo přihlášeno přes ORIS 112 účastníků z 19-ti různých klubů. Několik dalších jedinců se přihlásilo na místě závodu v den konání akce. Pořadatelé vytvořili 6 různých variant tratí, vypsanych kategorií bylo 11. Náhled trati hlavní mužské kategorie je k dispozici v příloze. Pohledem do přihlášek bylo zřejmé, že budou využity téměř všechny série závodních průkazů. Očekávání bylo velké, protože do té doby byly odzkoušeny jen 3 série.

Průběh závodu nenarušoval žádný velký kolaps, největším problémem bylo nakonec jen nevlídné počasí pro závodníky. Program fungoval spolehlivě, pár drobných maličností se našlo, ale ty neměly na zdárný průběh závodu vliv.



Obr. 14-1 závodník po vyčtení závodního průkazu

Všechny série závodních průkazů bez problému načítaly data. Změna rychlosti výpočtu a tisk mezičasů ze stále zvyšujícího počtu vyčtených závodníků nebyla znatelná. Ta by se mohla projevit až s větším množstvím závodníků, např. na českém poháru, kde počet startujících přesahuje 1000 osob. Všechny soubory, týkající se tohoto závodu, jsou k dispozici na přiloženém médiu a náhledy souborů jsou uvedeny v příloze. Výsledky po skončení závodu byly nahrány na ORIS. Odkaz se závodem lze najít na webové adrese <http://oris.orientacnisporty.cz/Zavod?id=3043>.

Výsledky závodu: OCAD.cz cup

Datum: 01.03.2015
Místo konání: Technická univerzita v Liberci
Pořadatel: TJN - OOB TJ Tatran Jablonec nad Nisou
Mapa:
 Technická univerzita v Liberci, 1 : 5 000, ekv. 2 m, stav 2/2015, autoři Jíří Daněk a Jan Pícek, vodovzdorně upravena
Ředitel závodu:
Hlavní rozhodčí:
Stavitel tratí:
Kategorie:
 D10 D14 D21K D21L D45 H10 H14 H21K H21L H45 P

Akce
Tisk
Jednoduchý tisk
Tisk do TXT
Export ČSOS
Export XML
Export XML 3.0
Stránka závodu

D10 0.80 km, 26 m, 9 kontrol

Mezičasy

MÍSTO	JMÉNO	REG. ČÍSLO	LICENCE	KLUB	ČAS	ZTRÁTA
1.	Matyášová Kristýna	CHA0555		OK Chrastava	9:09	
2.	Schröterová Adéla	CHA0551		OK Chrastava	11:33	+ 2:24
3.	Lamačová Barbora	BOR0750		OK Jiskra Nový Bor	13:20	+ 4:11
4.	Klustová Tereza	CHA0751		OK Chrastava	15:41	+ 6:32
MS.	Podrábská Bětko	VLI0451		Slavia Liberec orienteering	10:12	+ 1:03

Obr. 14-2 náhled výsledků na ORISu [11]

Po závodě jsem však narazil na jednu nepříjemnou věc, která se naštěstí neprojevila. Když jsem nahlédl do databáze, konkrétně do tabulky s daty ze závodních průkazů, všiml jsem si, že starší série (5. řada) má nastavený časový režim pouze na 12 hodin, na rozdíl od všech ostatních, které jsou nastaveny na 24 hodin. Důvody, proč se chyba neprojevila, jsou dva. První je takový, že závod startoval pomocí startovní krabičky, čili čas startu se načel ze stanice. Druhým důvodem bylo nepřesáhnutí hranice 12 hodin. Tento problém s časováním jsem později vyřešil.

15 ZÁVĚR

V první části této práce jsou teoreticky vysvětleny základní pojmy orientačních sportů. Popsán je vývoj kontrolních jednotek pro zaznamenávání průchodu závodníka. Ve zkratce je vysvětleno, s jakými typy závodů se lze setkat. Stručně jsou popsány některé podobné existující programy, jejich výhody a nevýhody. Jedna kapitola je věnována popisu systému ORIS.

Další část přísluší návrhu databáze. Zprvu jsou vylíčeny základní pojmy týkající se databází. Vysvětlený je typ zvoleného databázového systému, jeho specifika a způsob zápisu pro komunikování s daty. Stručně jsou popsány základní instrukce jazyka SQL, používané v této aplikaci. Přiložena je struktura databázového modelu, detailní popis atributů jednotlivých entit je přidán v příloze.

Hlavní část se zabývá vysvětlením jednotlivých částí programu, jak jdou za sebou při praktickém využití. Důkladně jsou popsány volby pro načtení závodníků do databáze při zakládání závodu (ORIS, textový soubor). Podrobně jsou rozepsány všechny funkce vložené v editaci záznamů. V kapitole o tratích jsou ujasněny volby pro jejich nahrání do programu. V části o tvorbě startovních časů jsou popsána specifika pro losování intervalových startů a samotná tvorba losování závodníků. Kapitola sběr dat z čipů se převážně zaobírá systémem SPORTident. Jednak je specifikována komunikace, typy závodních průkazů a jejich sběr dat. Taktéž je zde uvedeno zpracování získaných dat a jejich vytištění pro závodníka v cíli. Jedna z kapitol se věnuje exportovaným datům. Jsou zde definovány části a formáty dat, které lze exportovat. Náhled struktury programu je uveden v příloze.

Vytvořená aplikace byla otestována na veřejném závodě přímo v areálu Technické univerzity v Liberci. Testování proběhlo úspěšně bez výrazných komplikací. Celý průběh testování je popsán v kapitole č. 14. Soubory vztažené k této události jsou k dispozici na přiloženém médiu, náhledy souborů jsou k dispozici v příloze.

V průběhu tvorby programu se největší problémy vyskytovaly u závodních průkazů. Jednalo se hlavně o problémy staré série 5, která má oproti všem ostatním odlišné formátování času. Série 5 taktéž nedisponuje načítáním dat ze speciálních kontrol (clear, check). Pokusil jsem se problém této série vyřešit, byly provedeny příslušné testy. Nicméně, celá aplikace je poměrně rozsáhlá a v budoucnu budou potřeba provést další testy, které mohou odhalit další nedostatky.

Během tvorby na této práci jsem se naučil programovat v jazyce C#. Naučil jsem se nejen základní syntaxi zmíněného jazyka, ale i práci s událostmi, které jsou základem pro desktopové aplikace. Nedílnou znalostí byla práce s Windows Forms- frameworkem .NETu, umožňující jednoduchou práci s formulářovými prvky. Dalším přínosem bylo získání znalostí ohledně práce s databázemi a jazykem SQL. Z mého pohledu je jazyk C# velmi silný nástroj, se kterým se dají tvořit rozsáhlé aplikace „poměrně snadno“ i díky tomu, že na internetu je rozsáhlá informační podpora. Na druhou stranu nelze využít tyto aplikace na všech operačních systémech.

SEZNAM POUŽITÉ LITERATURY

Materiálové listy:

- [1] HUDDLESTON, J., AGARWAL, V., V.: *Databáze v C# 2008* Praha: Computer Press, 2009
- [2] NAGEL, CH., EVJEN, B., GLYNN J., SKINNER, M., WATSON, K.: *C# 2008 Programujeme profesionálně* Praha: Computer Press, 2013
- [3] SPORTident: *PC-Programmes Guide*, 2013

Elektronické zdroje informací:

- [4] C# Reference, [cit. 2015-04-12]. Dostupné z: <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx>
- [5] C# Programming Guide, [cit. 2015-04-12]. Dostupné z: <https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>
- [6] *Mapa pro OB* [online]. [cit. 1998-06-20]. Dostupné z: http://www.orientering-history.info/abc_ob.php
- [7] *Systém OB2000* [online]. [cit. 2012-02-10]. Dostupné z: <http://bestik.cz/vysledky/help.php?lang=en>
- [8] *Systém ORIS* [online]. [cit. 2013-01-01]. Dostupné z: <http://oris.orientacnisporty.cz/>
- [9] *SI jednotka pro sběr dat* [online]. [cit. 2010-07-15]. Dostupné z: <https://www.sportident.co.uk/equipment/catalogue/item.php?id=25>
- [10] *Závodní průkaz s SI jednotkou* [online]. [cit. 2014-10-05]. Dostupné z: <http://sportident-android.com/>
- [11] *Závod OCAD cup* [online]. [cit. 2015-03-02]. Dostupné z: <http://oris.orientacnisporty.cz/Zavod?id=3043>
- [12] *Specifikace čipových karet série 5 a 6* [online]. [cit. 2011-10-02]. Dostupné z: http://www.sportident.dk/Manualer/SPORTident-Cards-overview_en.pdf
- [13] *Specifikace čipových karet série 8 – 11* [online]. [cit. 2014-01-01]. Dostupné z: http://www.sportident.com/images/PDF/1_si_base_products/1_control%20cards/SPORTident-Cards-overview_cz.pdf

[14] *Specifikace čipových karet série pcard, tcard a fcard* [online]. [cit. 2014-01-01].

Dostupné z:

http://www.sportident.com/images/PDF/1_si_base_products/1_control%20cards/SPORTident-SmartCards-overview_en.pdf

[15] PICEK, J: *Ocad cup mapa s tratí kat. H21L*, 2015

SEZNAM PŘÍLOH

Příloha č. 1: Specifikace čipových karet série 5 a 6

Příloha č. 2: Specifikace čipových karet série 8-11

Příloha č. 3: Specifikace čipových karet série pcard, tcard a fcard

Příloha č. 4: OCAD cup – náhled přihlášek po kategorii H21L

Příloha č. 5: OCAD cup – náhled přihlášek po oddílech

Příloha č. 6: OCAD cup – náhled startovky po kategorii H21L

Příloha č. 7: OCAD cup – náhled startovek po oddílech

Příloha č. 8: OCAD cup – náhled konečných výsledků kategorie H21L

Příloha č. 9: OCAD cup – náhled mezičasů kategorie H21L

Příloha č. 10: OCAD cup – mapa s tratí pro kategorii H21L

Příloha č. 11: Popis atributů jednotlivých entit databáze

Příloha č. 12: Struktura aplikace

Příloha č. 13: Kompletní zdrojový kód (DVD)

Příloha č. 14: Soubory vztažené k OCAD cupu (DVD)

SPORTident - Cards

November, 2011



	<i>SI-Card5</i>	<i>SI-Card6</i>
	production terminated in 2011	production phase out in 2012
card number range	1 499999	500000 999999
card owner data files	no	complete address
life time	unlimited	unlimited
typical application	orienteering, adventure sports	adventure sports, orienteering
model type	finger stick	finger stick
size	65 mm * 17 mm	70 mm * 19 mm
colours	4 basic colours	2*9 basic colours,
data exchange time	330 ms	130 ms
time format	12h	1 week abs., 4 weeks relative
code number range	1...256	1...1024
total no. of data records	39	68
control records	30 + 6 (code no only)	64
special records	start finish	start finish clear check
SI-System compatibility Firmware	all SI-stations	all SI-stations
customer services		reservation service

Příloha č. 2: Specifikace čipových karet série 8-11 [13]

SPORTident-Čipy

V1.2 Leden 2014



	SIAC (Active Card)	SI čip 11	SI čip 10	SI čip 9	SI čip 8
Volba čísla v rozsahu	8 000 000 ... 8 999 999	9 000 000 ... 9 999 999	7 000 000 ... 7 999 999	1 000 000 ... 1 999 999	2 000 000 ... 2 999 999
Typické použití	Orientační běh, outdoorové aktivity	Orientační běh, outdoorové aktivity	Orientační běh, outdoorové aktivity	Orientační běh, outdoorové aktivity	Školní sporty, Orientační běh
Rychlost ražení	60 ms	60 ms	60 ms	115 ms	115 ms
Rychlost nulování	< 0,5 s	< 0,5 s	< 0,5 s	< 2 s	< 2 s
Frekvence RFID	125 kHz	125 kHz	125 kHz	125 kHz	125 kHz
Frekvence Rádio	2,4 Ghz (Short range radio)	—	—	—	—
Zpětnovazební signál	Optický (LED) a akustický	Optický (LED)	—	—	—
Počet mezipřechů	128	128	128	50	30
Další záznamy	Nulování, Kontrola, Start, Cíl	Nulování, Kontrola, Start, Cíl	Nulování, Kontrola, Start, Cíl	Nulování/Kontrola, Start, Cíl	Nulování/Kontrola, Start, Cíl
Formát dat	24 hod. formát, den v týdnu, počítadlo do 4 týdnů	24 hod. formát, den v týdnu, počítadlo do 4 týdnů	24 hod. formát, den v týdnu, počítadlo do 4 týdnů	24 hod. formát, den v týdnu, počítadlo do 4 týdnů	24 hod. formát, den v týdnu, počítadlo do 4 týdnů
Firmware čip	V 3.00 a vyšší	—	—	—	—
Kompatibilita s SI systémem + Firmware SI kontroly	BSx7/8: V 5.74 a vyšší V 2.14 a vyšší	BSx7/8: V 5.74 a vyšší	BSx7/8: V 5.74 a vyšší	BSx7/8: V 5.65 a vyšší	BSx7/8: V 5.65 a vyšší
Použití při teplotách	-20°C ... +50°C	-20°C ... +50°C	-20°C ... +50°C	-20°C ... +50°C	-20°C ... +50°C
Ochranná třída	IP 67 (DIN EN 60529) *	IP 67 (DIN EN 60529) *	IP 67 (DIN EN 60529) *	IP 67 (DIN EN 60529) *	IP 67 (DIN EN 60529) *
Životnost baterie	Až 4 let	Až 30 000 zápisů mezipřechů	—	—	—
Záruka	2 let s výjimkou baterií	5 let s výjimkou baterií	5 let	5 let	5 let
Hmotnost	11 g	10 g	8 g	8 g	8 g
Rozměry	68 mm x 21 mm x 12 mm	71 mm x 19 mm x 8 mm	71 mm x 19 mm x 8 mm	71 mm x 19 mm x 8 mm	71 mm x 19 mm x 8 mm
Barvy	4 základních barev pro tělo, bílý hrot s SI logem	Průhledné tělo, chromovaný hrot s SI logem	Průhledné tělo, různobarevná hlava	9 základních barev pro tělo nebo hrot	Tělo červené nebo modré, černý hrot
Zákaznický servis	—	Online rezervační servis	Online rezervační servis	Online rezervační servis, Možný požisk dle přání zákazníka	—

* IP 67 (DIN EN 60529) - Ochrana proti vniknutí vody při krátkodobém ponoření

SPORTident GmbH • Arnstadt • Germany • www.sportident.com • support@sportident.com

Příloha č. 3: Specifikace čipových karet pcard, tcard a fcard [14]

SPORTident-Cards – Overview

V1.1 January 2014



	<i>pCard</i>	<i>tCard</i>	<i>fCard</i>
Card number range	4 000 000 ... 4 999 999	6 000 000 ... 6 999 999	14 000 000 ... 14 999 999
Typical application	Outdoor sports, corporate activities	Tourism, permanent courses	Permanent control points (orienteeering)
Data exchange time	115 ms	115 ms	—
Clear time	< 2 s	No clear process needed (oldest data records will be overwritten)	—
Intermediate times	20	25	—
Special records	CLEAR, START, FINISH	START, FINISH	—
Time format	24h format, day of week, counter for up to 4 weeks	24h format, day of week, full calendar	—
SI-System compatibility + Firmware stations	BSx7/8: V 5.65 and higher BS11: V 2.01 and higher	BSx7/8: V 5.65 and higher BS11: V 2.01 and higher	BSx7/8: V 5.80 and higher
Operating range	-20°C ... +50°C	-20°C ... +50°C	-20°C ... +50°C
Warranty	2 years	2 years	2 years
Weight	4 g	4 g	4 g
Dimensions	54 mm x 54 mm x 0.8 mm	54 mm x 54 mm x 0.8 mm	54 mm x 54 mm x 0.8 mm
Colours	Free designable full colour imprint on both sides	Free designable full colour imprint on both sides	Free designable full colour imprint on both sides
Customer services	Custom specific imprint	Custom specific imprint	Custom specific imprint

Příloha č. 4: OCAD cup – náhled přihlášek po kategorii H21L

Přihlášky po kategoriích

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 9

3. 3. 2015 11:50:19

H21L (29) - 2,4km - 80m - 18K

1	Jíran	Jan	GHOST RUBENA RACING TEAM	XHK7701	681855	C
2	Chmelař	Lukáš	GHOST RUBENA RACING TEAM	XHK9100	7003041	
3	Hošek	Radim	KOB DOBRUŠKA	DOB8737	301627	C
4	Pešík	Michal	NEREGISTROVANÍ	NNN2501	2025338	
5	Keltner	Marek	NEREGISTROVANÍ	NNN2502	9201066	
6	Severýn	Otto	NEREGISTROVANÍ	NNN2506	2067951	
7	Dušek	Martin	OB T. J. SOKOL ŽAMBERK	ZAM9602	436957	B
8	Kolín	Jan	ODDÍL OB KOTLÁŘKA	DKP7701	9007701	C
9	Janata	Martin	ODDÍL OB KOTLÁŘKA	DKP8306	888608	R
10	Nývlt	Martin	ODDÍL OB KOTLÁŘKA	DKP8200	1207407	C
11	Pospíšil	Ondřej	ODDÍL OB KOTLÁŘKA	DKP8717	7202874	C
12	Lederer	Pavel	OK DOKSY	DOK6600	8147	C
13	Polák	Jaroslav	OK CHRASTAVA	CHA7500	7201775	C
14	Tomanica	Vojtěch	OK CHRASTAVA	CHA7601	8167	
15	Burda	Matěj	OK JISKRA NOVÝ BOR	BOR9608	7667667	A
16	Kořínek	Tomáš	OK JISKRA NOVÝ BOR	BOR9007	7003001	R
17	Eiselt	Jan	OOB TJ LOKOMOTIVA TEPLICE	LTP8801	682092	C
18	Štembera	Jan	OOB TJ TATRAN JABLONEC NAD NISOU	TJN7513	51536	C
19	Váňa	Martin	OOB TJ TURNOV	TUR8401	1840303	C
20	Wolf	Daniel	OOB TJ TURNOV	TUR9100	515499	R
21	Schulhof	Jan	OOB TJ TURNOV	TUR8900	1969696	C
22	Luštický	Martin	OOB TJ TURNOV	TUR8700	2035709	R
23	Kolaja	Jan	SLAVIA LIBEREC ORIENTEERING	VLI8100	2067313	C
24	Petr	Pavel	SLAVIA LIBEREC ORIENTEERING	VLI9400	8288	C
25	Podrábský	Ctibor	SLAVIA LIBEREC ORIENTEERING	VLI7001	2058472	C
26	Martan	Jiří	SLAVIA LIBEREC ORIENTEERING	VLI7300	8129	C
27	Mokrý	Pavel	SLAVIA LIBEREC ORIENTEERING	VLI7501	2058479	
28	Pavlíček	Lubomír	SOOB TJ SOKOL KRALUPY N.VLT.	KPY7800	1470170	C
29		Vakant	VAKANT	VAKH21L+1		

Příloha č. 5: OCAD cup – náhled přihlášek po oddílech

Přihlášky po oddílech

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 1

3. 3. 2015 11:50:26

OK JISKRA NOVÝ BOR (13)

1	Lamačová	Barbora	D10	BOR0750	4124	C
2	Tymešová	Anna	D14	BOR0253	4408039	C
3	Kameníková	Kateřina	D21K	BOR7758	8441	C
4	Tučková	Monika	D21K	BOR7650	7913579	C
5	Burdová	Adéla	D21L	BOR9161	7434894	C
6	Šáková	Michaela	D21L	BOR8850	8481	C
7	Folbrechtová	Šárka	D21L	BOR9051	1602628	C
8	Kameníková	Andrea	D21L	BOR9751	2051536	A
9	Burda	Matěj	H21L	BOR9608	7667667	A
10	Kořínek	Tomáš	H21L	BOR9007	7003001	R
11	Pavlovec	Petr	H45	BOR6303	219203	C
12	Votýpková	Zuzana	P	BOR7958	4408037	C
13	Lamač	Vítek	H10	BOR1100	8550	C

Příloha č. 6: OCAD cup – náhled startovky po kategorii H21L

Startovky po kategoriích

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 9

3. 3. 2015 11:49:07

H21L (29) - 2,4km - 80m - 18K

1	Burda	Matěj	BOR9608	7667667	A OK JISKRA NOVÝ BOR	0:00
2	Kolín	Jan	DKP7701	9007701	C ODDÍL OB KOTLÁRKA	0:00
3	Janata	Martin	DKP8306	888608	R ODDÍL OB KOTLÁRKA	0:00
4	Nývlt	Martin	DKP8200	1207407	C ODDÍL OB KOTLÁRKA	0:00
5	Pavlíček	Lubomír	KPY7800	1470170	C SOOB TJ SOKOL KRALUPY N.VLT.	0:00
6	Lederer	Pavel	DOK6600	8147	C OK DOKSY	0:00
7	Kolaja	Jan	VLI8100	2067313	C SLAVIA LIBEREC ORIENTEERING	0:00
8	Petr	Pavel	VLI9400	8288	C SLAVIA LIBEREC ORIENTEERING	0:00
9	Polák	Jaroslav	CHA7500	7201775	C OK CHRSTAVA	0:00
10	Kořínek	Tomáš	BOR9007	7003001	R OK JISKRA NOVÝ BOR	0:00
11	Pospíšil	Ondřej	DKP8717	7202874	C ODDÍL OB KOTLÁRKA	0:00
12	Podrábský	Ctibor	VLI7001	2058472	C SLAVIA LIBEREC ORIENTEERING	0:00
13	Hošek	Radim	DOB8737	301627	C KOB DOBRUŠKA	0:00
14	Jíran	Jan	XHK7701	681855	C GHOST RUBENA RACING TEAM	0:00
15	Chmelař	Lukáš	XHK9100	7003041	GHOST RUBENA RACING TEAM	0:00
16	Váňa	Martin	TUR8401	1840303	C OOB TJ TURNOV	0:00
17	Martan	Jiří	VLI7300	8129	C SLAVIA LIBEREC ORIENTEERING	0:00
18	Eiselt	Jan	LTP8801	682092	C OOB TJ LOKOMOTIVA TEPLICE	0:00
19	Štembera	Jan	TJN7513	51536	C OOB TJ TATRAN JABLONEC NAD NISOU	0:00
20	Wolf	Daniel	TUR9100	515499	R OOB TJ TURNOV	0:00
21	Pešík	Michal	NNN2501	2025338	NEREGISTROVANÍ	0:00
22	Keltner	Marek	NNN2502	9201066	NEREGISTROVANÍ	0:00
23	Mokrný	Pavel	VLI7501	2058479	SLAVIA LIBEREC ORIENTEERING	0:00
24	Schulhof	Jan	TUR8900	1969696	C OOB TJ TURNOV	0:00
25	Severýn	Otto	NNN2506	2067951	NEREGISTROVANÍ	0:00
26	Dušek	Martin	ZAM9602	436957	B OB T.J. SOKOL ŽAMBERK	0:00
27	Luštický	Martin	TUR8700	2035709	R OOB TJ TURNOV	0:00
28	Vakant		VAKH21L+1		VAKANT	0:00
29	Tomanica	Vojtěch	CHA7601	8167	OK CHRSTAVA	0:00

Příloha č. 7: OCAD cup – náhled startovek po oddílech

Startovky po oddílech

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 1

3. 3. 2015 11:49:23

OK JISKRA NOVÝ BOR (13)

1	Lamačová	Barbora	BOR0750	4124	C D10	0:00
2	Tymešová	Anna	BOR0253	4408039	C D14	0:00
3	Kameníková	Kateřina	BOR7758	8441	C D21K	0:00
4	Tučková	Monika	BOR7650	7913579	C D21K	0:00
5	Burdová	Adéla	BOR9161	7434894	C D21L	0:00
6	Šáková	Michaela	BOR8850	8481	C D21L	0:00
7	Folbrechtová	Šárka	BOR9051	1602628	C D21L	0:00
8	Kameníková	Andrea	BOR9751	2051536	A D21L	0:00
9	Burda	Matěj	BOR9608	7667667	A H21L	0:00
10	Kořínek	Tomáš	BOR9007	7003001	R H21L	0:00
11	Pavlovec	Petr	BOR6303	219203	C H45	0:00
12	Votýpková	Zuzana	BOR7958	4408037	C P	0:00
13	Lamač	Vítek	BOR1100	8550	C H10	0:00

Příloha č. 8: OCAD cup – náhled konečných výsledků kategorie H21L

Konečné výsledky

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 9

3. 3. 2015 11:49:42

H21L (28/28) - 2,4km - 80m - 18K

1	Wolf	Daniel	OOB TJ TURNOV	TUR9100	R	14:31	+	00:00
2	Luštický	Martin	OOB TJ TURNOV	TUR8700	R	14:33	+	00:02
3	Nývlt	Martin	ODDÍL OB KOTLÁŘKA	DKP8200	C	14:38	+	00:07
4	Pospišil	Ondřej	ODDÍL OB KOTLÁŘKA	DKP8717	C	14:51	+	00:20
5	Schulhof	Jan	OOB TJ TURNOV	TUR8900	C	14:58	+	00:27
6	Váňa	Martin	OOB TJ TURNOV	TUR8401	C	15:12	+	00:41
7	Kolín	Jan	ODDÍL OB KOTLÁŘKA	DKP7701	C	15:42	+	01:11
8	Hošek	Radim	KOB DOBRUŠKA	DOB8737	C	15:55	+	01:24
9	Martan	Jiří	SLAVIA LIBEREC ORIENTEERING	VLI7300	C	16:30	+	01:59
10	Burda	Matěj	OK JISKRA NOVÝ BOR	BOR9608	A	16:49	+	02:18
11	Kolaja	Jan	SLAVIA LIBEREC ORIENTEERING	VLI8100	C	17:05	+	02:34
12	Podrábský	Ctibor	SLAVIA LIBEREC ORIENTEERING	VLI7001	C	17:44	+	03:13
13	Petr	Pavel	SLAVIA LIBEREC ORIENTEERING	VLI9400	C	17:49	+	03:18
14	Pavliček	Lubomír	SOOB TJ SOKOL KRALUPY N.VLT.	KPY7800	C	18:05	+	03:34
15	Polák	Jaroslav	OK CHRASTAVA	CHA7500	C	18:11	+	03:40
16	Severýn	Otto	NEREGISTROVANÍ	NNN2506		18:30	+	03:59
17	Eiselt	Jan	OOB TJ LOKOMOTIVA TEPLICE	LTP8801	C	18:57	+	04:26
18	Mokrý	Pavel	SLAVIA LIBEREC ORIENTEERING	VLI7501		19:00	+	04:29
19	Keltner	Marek	NEREGISTROVANÍ	NNN2502		19:09	+	04:38
20	Pešík	Michal	NEREGISTROVANÍ	NNN2501		19:44	+	05:13
21	Štembera	Jan	OOB TJ TATRAN JABLONEC NAD NISOU	TJN7513	C	23:05	+	08:34
22	Tomanica	Vojtěch	OK CHRASTAVA	CHA7601		27:33	+	13:02
	Janata	Martin	ODDÍL OB KOTLÁŘKA	DKP8306	R	DNS		
	Lederer	Pavel	OK DOKSY	DOK6600	C	DNS		
	Kořínek	Tomáš	OK JISKRA NOVÝ BOR	BOR9007	R	DNS		
	Jíran	Jan	GHOST RUBENA RACING TEAM	XHK7701	C	DNS		
	Chmelař	Lukáš	GHOST RUBENA RACING TEAM	XHK9100		DNS		
	Dušek	Martin	OB T.J. SOKOL ŽAMBERK	ZAM9602	B	DNS		

Příloha č. 9: OCAD cup – náhled mezičasů kategorie H21L

Mezičasy

OCAD.cz cup
kobriS © Tomáš Kobr

Stránka 14

3. 3. 2015 11:49:48

H21L - 2,4km - 80m - 18K

	1 (70)	2 (71)	3 (72)	4 (73)	5 (74)	6 (75)	7 (72)	8 (77)	9 (78)	10(79)
	11(81)	12(80)	13(82)	14(84)	15(87)	16(88)	17(83)	18(100)	19(F1)	
1 Wolf	00:26 (2)	00:50 (3)	01:02 (3)	00:29 (2)	00:26 (1)	00:25 (1)	00:19 (7)	00:53 (3)	01:02 (3)	01:35 (6)
OOB TJ TURNOV	00:26 (2)	01:16 (3)	02:18 (2)	02:47 (2)	03:13 (2)	03:38 (2)	03:57 (2)	04:50 (2)	05:52 (2)	07:27 (3)
	00:40 (4)	02:00 (6)	01:18 (6)	00:09 (2)	01:06 (1)	00:33 (2)	00:19 (2)	00:50 (7)	00:09 (5)	
	08:07 (3)	10:07 (2)	11:25 (3)	11:34 (3)	12:40 (1)	13:13 (1)	13:32 (1)	14:22 (1)	14:31 (1)	
14:31										
Daniel										
TUR9100										
2 Luštický	00:30 (8)	00:57 (7)	01:01 (2)	00:28 (1)	00:26 (1)	00:29 (8)	00:16 (1)	00:52 (2)	00:58 (2)	01:28 (2)
OOB TJ TURNOV	00:30 (8)	01:27 (9)	02:28 (6)	02:56 (5)	03:22 (5)	03:51 (5)	04:07 (4)	04:59 (4)	05:57 (3)	07:25 (2)
	00:36 (1)	02:08 (8)	01:14 (3)	00:08 (1)	01:09 (5)	00:43 (13)	00:18 (1)	00:42 (1)	00:10 (6)	
	08:01 (2)	10:09 (3)	11:23 (2)	11:31 (2)	12:40 (1)	13:23 (2)	13:41 (2)	14:23 (2)	14:33 (2)	
14:33										
Martin										
TUR8700										
3 Nývlt	00:24 (1)	00:49 (2)	01:06 (6)	00:29 (2)	00:26 (1)	00:28 (5)	00:19 (7)	00:53 (3)	01:18 (20)	01:30 (3)
ODDÍL OB KOTLÁRKA	00:24 (1)	01:13 (1)	02:19 (4)	02:48 (3)	03:14 (3)	03:42 (3)	04:01 (3)	04:54 (3)	06:12 (5)	07:42 (5)
	00:37 (2)	01:55 (2)	01:13 (2)	00:11 (8)	01:10 (6)	00:35 (4)	00:20 (3)	00:47 (2)	00:08 (2)	
	08:19 (4)	10:14 (4)	11:27 (4)	11:38 (4)	12:48 (3)	13:23 (2)	13:43 (3)	14:30 (3)	14:38 (3)	
14:38										
Martin										
DKP8200										
4 Pospíšil	00:27 (3)	00:51 (4)	01:00 (1)	00:31 (7)	00:31 (9)	00:28 (5)	00:19 (7)	00:57 (6)	01:04 (5)	01:33 (4)
ODDÍL OB KOTLÁRKA	00:27 (3)	01:18 (4)	02:18 (2)	02:49 (4)	03:20 (4)	03:48 (4)	04:07 (4)	05:04 (5)	06:08 (4)	07:41 (4)
	00:46 (9)	01:59 (4)	01:15 (4)	00:15 (17)	01:06 (1)	00:33 (2)	00:21 (6)	00:49 (4)	00:06 (1)	
	08:27 (5)	10:26 (5)	11:41 (5)	11:56 (5)	13:02 (4)	13:35 (4)	13:56 (4)	14:45 (4)	14:51 (4)	
14:51										
Ondřej										
DKP8717										
5 Schulhof	00:27 (3)	00:48 (1)	01:02 (3)	00:29 (2)	00:26 (1)	00:25 (1)	00:16 (1)	00:50 (1)	00:57 (1)	01:24 (1)
OOB TJ TURNOV	00:27 (3)	01:15 (2)	02:17 (1)	02:46 (1)	03:12 (1)	03:37 (1)	03:53 (1)	04:43 (1)	05:40 (1)	07:04 (1)
	00:41 (6)	01:53 (1)	01:09 (1)	00:28 (20)	01:49 (20)	00:32 (1)	00:25 (11)	00:47 (2)	00:10 (6)	
	07:45 (1)	09:38 (1)	10:47 (1)	11:15 (1)	13:04 (5)	13:36 (5)	14:01 (5)	14:48 (5)	14:58 (5)	

Technická univerzita v Liberci

měřítko 1 : 5 000
ekvidistance 2m
stav únor 2015

Evidence map: 15-05-02 V
Mapovali a kreslili v měřítku 1 : 2 000: Jiří Daněk, Jan Pícek
Hlavní kartograf: Jiří Daněk
Mapové podklady: DMR 5G, ortofoto
Zpracování: www.ocad.cz
Vydal: TJ Tatran Jablonec
Tisk: Tiskárna macek-kusala, Jablonec nad Nisou
Správce mapy: Jiří Daněk, Gorkého 619/17, Liberec 460 01

OCAD®
the smart software
for cartography
www.ocad.cz

H21L

OCAD.cz sprint / 1.3.2015		2,4 km	80 m
H21L			
1	70	■	□
2	71	■	□
3	72	■	□
4	73	■	□
5	74	■	□
6	75	■	□
7	72	■	□
8	77	■	□
9	78	■	□
10	79	■	□
11	81	■	□
12	80	■	□
13	82	■	□
14	84	■	□
15	87	■	□
16	88	■	□
17	83	■	□
18	100	■	□

R1	R2	R3
----	----	----



MjMaps
www.mjmaps.cz

Příloha č. 11: Popis atributů jednotlivých entit databáze

Entita ZÁVOD:

- závod ID - název závodu, primární klíč
- organizátor - pořadající orgán
- odvětví - forma závodu
- místo - působiště probíhající akce
- název oris - název použitý pro export výsledků na webový server
- kód závodu - kód použitý pro export výsledků na webový server

Entita ETAPA:

- etapa ID - označení etapy, primární klíč
- start - startovní čas etapy
- datum - termín etapy
- závod ID - zděděný atribut od entity závod

Entita TRATĚ:

- trať ID - název trati, primární klíč
- délka - délka trati v kilometrech
- převýšení - minimální vertikální převýšení trati
- počet kontrol - suma kontrol na trati
- etapa ID - zděděný atribut od entity etapa

Entita SEKVENCE KONTROL:

- pořadové č. - pozice kontroly na trati, primární klíč
- hnízdo - speciální označení pro kontrolu, vyskytuje-li se v určité skupině kontrol
- trať ID - zděděný atribut od entity trať
- kontrola ID - zděděný atribut od entity kontrola

Entita KONTROLA:

- kontrola ID - označení kontroly, primární klíč

Příloha č. 11: Popis atributů jednotlivých entit databáze

Entita KATEGORIE:

kategorie ID	- pojmenování kategorie, primární klíč
startovní interval	- čas startovních rozestupů mezi jednotlivými závodníky
časový limit	- limit pro absolvování trati
start	- čas odstartování prvního účastníka dané kategorie
počet startujících	- počet účastníků v kategorii
počet vakantů	- počet vakantů v kategorii
termín 1	- vklad závodníka v prvním termínu přihlášek
termín 2	- vklad závodníka v druhém termínu přihlášek
termín 3	- vklad závodníka ve třetím termínu přihlášek
typ startu	- forma startování účastníků (intervalový, hromadný, startovní krabíčka)
typ závodu	- forma závodu (obecný, kvalifikace)
nadřazená trať	- jméno nadřazené trati, použití v kvalifikacích, pro utajení pod-kategorie.
řazení vakantů	- jedná se o zařazení do startovních listin, do které části startovky mají být umístěny vakanti
losováno	- stav, jestli je startovní listina losována
etapa ID	- zděděný atribut od entity etapa

Entita ZÁVODNÍK:

reg. ID	- registrační číslo, primární klíč
jméno	- křesní jméno sportovce
příjmení	- příjmení sportovce
číslo čipu	- identifikační kód závodního průkazu
licence	- obdržená licence pro daný rok
startovní číslo	- startovní číslo závodníka
poznámka	- požadavky účastníka v přihlášce
půjčit čip	- stav o půjčení závodního průkazu pořadatelem
termín	- termín přihlášky
brzký start	- start vepředu startovní listiny
pozdní start	- start na konci startovní listiny
start	- přidělený start pořadatelem

Příloha č. 11: Popis atributů jednotlivých entit databáze

ms	- stav o závodění mimo soutěž
klub ID	- zděděný atribut od entity klub
kategorie ID	- zděděný atribut od entity kategorie

Entita KLUB:

klub ID	- jméno oddílu, primární klíč
zkratka	- zkratka oddílu daná svazem ČSOS
suma čipy	- celková cena, za půjčení čipů
suma závodníci	- celková cena, za startovné závodníků v oddíle
suma ostatní	- celková cena, za ostatní služby
platba	- platba oddílu

Entita DATA ČIP:

clear code	- kód kontroly pro mazání předchozích dat
clear time	- čas nulovací kontroly, zaznamenaný při jejím průchodu
check code	- kód ověřovací kontroly (ověření úspěšného vynulování čipu)
check time	- čas ověřovací kontroly, zaznamenaný při jejím průchodu
start code	- kód startovní kontroly
start time	- čas startu
start time s. k.	- čas startu ze startovní kontroly
finish code	- kód cílové kontroly
finish time	- čas cílové kontroly, zaznamenaný při jejím průchodu
cip	- číslo závodního průkazu
chyby	- souhrn chyb, při vyčítání čipu (oddělené středníkem)
kódy	- souhrn všech kódů kontrol, kterými závodník prošel
časy	- souhrn všech časů kontrol, kterými závodník prošel
mezičasy	- seznam kódů na trati s mezičasem mezi kontrolami a celkovým mezičasem od startu závodu
zbytek kontrol	- seznam kódů s časem, jenž závodník orazil navíc mimo danou trať
celkový čas	- konečný čas závodníka
opakování	- stav, je-li čip opakovaně vyčten
disk	- stav, je-li závodník diskvalifikován

Příloha č. 12: Struktura aplikace

