MKS_JG - Instrukcja użytkownika

Autor: Mateusz Lubecki SP8EBC

Bielsko-Biała
Marzec - Kwiecień 2018

Oprogramowanie MKS_JG dedykuje trenerowi Przemkowi Pochłódowi jako wyraz mej wdzięczności za poświecany mi czas, jak i ogrom pracy wkładany przez Niego w rozwoj sportowy dzieciaków ze Szkoly Mistrzostwa Sportowego w Karpaczu. Składam przy tym slowa uznania dla Tomka Koćmierowskiego, Marty Pochłód, Moniki Pawlak i pozostalych nieznanych mi zapewne z imienia i nazwiska trenerów pracujących w SMS Karpacz, Międzyszkolnym Klubie Sportowym Jelenia Góra i KS Śnieżka Karpacz.

Autor

Spis treści

Rozdział 1

Przedmowa

Pomysł na stworzenie oprogramowania takiego jak MKS_JG zrodził się u mnie początkiem marca 2018 roku, kiedy przyglądałem się Międzywojewódzkim Mistrzostwom Młodzików zorganizowanym na torze sankowym w Karpaczu przy ulicy Świętokrzyskiej, który został na tę okoliczność wylodzony na odcinku około stu metrów. Zwróciłem uwagę na fakt, że używane przez klub urządzenia do pomiaru czasu ślizgu wydają się posiadać możliwość współpracy z komputerem przez obecny w 'centralce pomiarowej' port szeregowy w typowym standardzie RS232. Stworzenie współpracującego z nimi, szytego na miarę oprogramowania znacznie ułatwiało by przeprowadzanie zawodów oraz treningów, jak również odciążało by organizatorów od nie ma co ukrywać żmudnego i narażonego na omyłki wpisywania odczytanych ręcznie czasów do arkusza kalkulacyjnego. Dedykowane oprogramowanie daje zawsze możliwość w zasadzie dowolego rozbudowywania funkcjonalności, dlatego i tutaj bardzo szybko pojawiły się pomysły na kolejne typowo „sankowe" opcje. Zaczynając od generowania raportów podobnych do tych, które tworzą systemy pomiaru na torach klasy olimpijskiej (Koenigsee, Innsbruck-Iglis, Sigulda itp) idąc przez prowadzenie 'długofalowych' statystyk z większej ilości treningów i zawodów, a kończąc na obsłudze sztafety sankowej.

Dla mnie jako autora MKS_JG jest to z jednej strony pewien swoisty wkład w rozwój MKS Sporty Zimowe Jelenia Góra, z drugiej strony forma odwdzięczenia się trenerowi Przemkowi Pochłódowi za czas który mi poświęca i wiedzę którą mi przekazuję. Z trzeciej zaś strony jest to trening umiejętności programowania w technologii Java, który być może kiedyś ułatwi zdobycie nowej pracy tak samo jak w 2017 roku uczyniły to moje stacje pogodowe, rozsiane po całej południowej Polsce.

Niniejsza instrukcja obsługi jest przeznaczona dla użytkownika końcowego i ma na celu przedstawienie oraz wyjaśnienie obsługi programu MKS_JG w możliwie jak najprostszy sposób. Aby ułatwić przekaz złamana została jedna z podstawowych zasad pisania dokumentacji technicznej zakazująca pisania tekstu w pierwszej osobie, co w tym dokumencie będzie miało miejsce. Gdzieniegdzie będę się starał wplatać w tekst pewne szczeguły techniczne, które choć nie niezbędne do sprawnego posługiwania się oprogramowaniem, rzucą nieco światła na kulisy jego działania oraz powstawania, mogą też być dla niektórych po prostu ciekawe.

Jako odskocznię od przydługawego i niezbyt pasjonującego tekstu technicznego będę się starał zamieszczać nieco drętwego humoru paralotniowego i programistycznego w formie krótkich „sucharów", czy też różnego rodzaju przypowieści i innych mądrości ludowych. Jeżeli szanowny czytelniku nie będziesz ich rozumiał, bądź będą Ci się wydawały kompletnie nieśmieszne to nie ma powodu do jakichkolwiek obaw. W sumie to nawet dobrze, bo ani programiści ani paralotniarze to nie są do końca normalni ludzie...

Rozdział 2

Postanowienia licencyjne

Oprogramowanie MKS_JG powstaje niejako w czynie społecznym dlatego naturalnym jest, że będzie dostępne dla użytkownika końcowego za darmo na czas nieoznaczony. MKS_JG jak sama nazwa wskazuje powstaje głównie z myślą o Międzyszkolnym Klubie Sportowym Jelenia Góra, oraz jednocześnie Szkole Mistrzostwa Sportowego z Karpacza, jeżeli jednak zajdzie taka potrzeba może być wykorzystywany również przez dowolny inny klub sanokowy jak np. UKS Nowiny Wielkie, KS Śnieżka Karpacz itp. Aplikacja jest silnie zorientowana na saneczkarstwo i bez gruntownych zmian w kodzie źródłowym raczej nie znajdzie zastosowania poza tym sportem.

Z formalnego punktu widzenia aplikacja w formie binarnej, oraz jej kod źródłowy o ile nie zostało to w jawny sposób zadeklarowane inaczej jest dystrybuowana na licencji **GNU General Public License version 3** przedstawionej przez Free Software Foundation. W ogólności użycie tejże licencji (zwanej dalej jako „GPL") niesie ze sobą konsekwencję takie jak:

- Aplikacja może być dowolnie dystrybuowana i użytkowana do celów niekomercyjnych oraz komercyjnych, zarówno w formie kodów źródłowych jak i skompilowanej (tj. gotowej do uruchomienia) wersji binarnej.
- Kod źródłowy aplikacji może być dowolnie modyfikowany przez użytkownika **pod warunkiem, że wprowadzone do kodu modyfikację zostaną upublicznione.**
- Użycie licencjonowanego na zasadach GPL kodu aplikacji MKS_JG w innym oprogramowaniu automatycznie podowuje, że oprogramowanie to również musi być licencjonowane na zasadach GPL.

Użycie GPL jako metody licencjonowania było jednym z filarów dużego sukcesu i rozwoju systemu Linux, gdyż deweloperzy (programiści) tworzący np. kernel czyli jądro systemu operacyjnego mogli mieć pewność, że ich charytatywna praca nie będzie zawłaszczona bez żadnego wkładu w projekt przez firmy tworzące komeryjne oprogramowanie. Dzięki temu mamy obecnie np. Android, który jest de facto jedną z wielu dystrybucji systemu Linux, oraz Internet jako taki który w swoich kulisach bazuje głównie na sprzęcie i oprogramowaniu serwerowym wykorzystującym/działającym pod kontrolą różnych dystrybucji systemu Linux.

Na chwilę obecną wyjątkiem od licencjonowania GPL jest komponent odpowiedzialny za obsługę komunikacji z systemem pomiaru czasu firmy Sectro z Jeleniej Góry, czyli klasy SectroGateTimeData oraz SectroParser znajdujące się w paczce pl.jeleniagora.mks.chrono . Protokół komunikacyjny firmy Sectro stanowi jej tajemnicę przemysłową i bez uprzedniej zgody nie będzie publikowany.

Rozdział 3

Wymagania funkcjonalne

"Walking on water and developing software from a specification are easy if both are frozen." W tłumaczeniu *„Chodzenie po wodzie i tworzenie oprogramowania na podstawie wymagań jest łatwe jeżeli obydwie rzeczy są zamrożone"*, cytat z Essays on object-oriented software engineering autor Edward V. Berard

Rozdział ten nie stanowi de facto instrukcji obsługi a jedynie spis funkcjonalności (fachowo nazywa się to Functional Requirements Document - FRD), które MKS_JG powinien posiadać wraz z priorytetem ich wdrożenia. Jeżeli szukasz konkretnych informacji i wskazówek na temat użytkowania MKS_JG przeskocz do kolejnych rozdziałów.

W każdym projekcie programistycznym podstawą do jakiejkolwiek pracy są wymagania, spisane nawet w najbardziej ogólnej formie na podstawie konkretnych wymagań klienckich. W dużych projektach jest to niemalże obowiązkowy punkt od którego wszystko się zaczyna. Istnieją nawet specjalne, bardzo duże programy służące tylko i wyłącznie do zarządzania oraz katalogowania tychże wymagań. Jednym z nich jest np. serdecznie znienawidzone przeze mnie IBM Rational DOORS (Dynamic Object Oriented Requiremenets System). MKS_JG jest oczywiście dość skromnym projektem zarządzanym, rozwijanym i testowanym jednosobowo przezeze mnie, aczkolwiek nawet tutaj w pewnym momencie wymagania funkcjonalne muszą zostać jasno określone.

Jako, że program rozwijam w wolnym czasie muszę ustalić sam sobie priorytety i wskazać która funkcjonalność musi zostać wdrożona i przetestowana w pierwszej kolejności. Dodatkowo jasne wyartykułowanie pozwalana na zapobieżenie sytuacji w której wymagania będą naginane do twardej rzeczywistości, w której nieubłaganie będzie pojawiało się mnóstwo problemów technicznych, bugów, awarii itp ;) Wspomniane priorytety zostały podzielone na cztery kategorie:

HP Najwyższy priorytet okreslający funkcjonalności, które stanowią podstawę działania programu przez co muszą być implementowane w pierwszej kolejności i przy tym solidnie przetestowane.

MP Średni priorytet określa równie ważne funkcjonalności jednak wdrażane za tymi oznaczonymi jako HP, głównie ze względu na wzajemną interakcję między modułami (MP wykorzystuje HP). Na MP poświęca się mniejszy nakład pracy na implementację i testy.

LP Funkcjonalności najniższego priorytetu. Stanowiące istotny element oprogramowania ale jednak nie na tyle aby utraciło ono bez nich swoją podstawową rolę. Są zaplanowane do implementacji i testów do pierwszej wersji stabilnej, ale w przypadku poślizgów czasowych mogą zostać wyrzucone całkowicie z projektu bądź przeniesione na kolejną wersję stabilną

FR Są to elementy architektury programu, które zostały z góry przeniesione na kolejne wydania wersji stabilnej (FR - Further Releases). Są to funkcjonalności nie tyle nieistotne ale stanowiące albo rozwinięcie tego co jest implementowane w ramach pozostałych priorytetów, albo będące znacznym elementem wymagającym bardzo dużego nakładu pracy, nie stanowiąc jednocześnie istoty działania programu.

Jako deadline, czyli datę wypuszczenia pierwszej w pełni gotowej do pracy wersji MKS_JG przyjąłem 15 (a w zasadzie 14) września 2018, czyli termin sankorolkowego Memoriału im. Mariusza Warzyboka w Karpaczu.

Pierwszym podstawowym wymaganiem funkcjonalnym oprogramowania MKS_JG jest kompleksowa obsługa klubu sankowego zarówno pod kątem organizacji zawodów z rejestracją czasu ślizgu, tworzeniem list rankingowych, raportów itp, jak również prowadzenia długofalowych statystyk dla każdego zawodnika.

Dalsze wymagania funkcjonalne przedstawiłem na poniższych listach i wypunktowaniach. Kolejność jest w zasadzie dowolna, choć jest pogrupowana pod względem „ramowych funkcjonalności" typu interfejs użytkownika, format zapisu pliku itp.

1. Interfejs użytkownika
   (a) HP Oprogramowanie MKS_JG powinno posiadać graficzny interfejs użytkownika (GUI) obsługiwany zarówno przy pomocy klawiatury i myszy.
   (b) HP Interfejs graficzny użytkownika powinien być zaprojektowany w sposób który umożliwia obsługę jedynie przy pomocy klawiatury bez używania myszy, oraz tylko przy pomocy myszy przy założeniu używania klawiatury ekranowej.
   (c) LP Interfejs powinien umożliwiać zmianę języka, przy czym dostępne powinny być conajmniej cztery:
      i. Polski
      ii. Dialekt / gwara Górnośląska
      iii. Czeski
      iv. Angielski
   (d) HP Interfejs użytkownika powinien zostać podzielony na kilka dużych bloków (modułów) funkcjonalnych mających postać osobnych okien roboczych wyświetlających interfejs każdego modułu z osobna. Dostępne mają być conajmniej takie jak:
      i. EKRAN GŁÓWNY - Możliwe jak najprostszy, służący do wyboru moduły funkcjonalnego
      ii. MENADŻER ZAWODNIKÓW - Moduł służący do podglądu, edycji, dodawania oraz usuwania zawodników i ich danych takich jak Imię, Nazwisko, email, przynależność klubowa. Moduł musi posiadać również możliwość przeglądu statystyk sankowych dla każdego zawodnika z osobna, oraz umożliwiać eksport tych statystych w formie raportu.
      iii. OBSŁUGA ZAWODÓW/TRENINGÓW - Moduł służący do kompleksowej obsługi zawodów i treningów saneczkowych w oparciu o ręcznie wprowadzane czasy ślizgów, jak i te otrzymane automatycznie z chronometru. Więcej wymagań funkcjonalnych w punkcie...
   (e) HP Interfejs użytkownika musi być maksymalnie uproszczony. W oknach intefejsu powinna znajdować się minimalna potrzebna ilość opcji tak aby użytkownik nie odniósł wrażenia, że program jest przeładowany funkcjonalnością i jednocześnie trudny

w obsłudzę. Zasada ta odnosi się w szczególności do modułu „Obsługa zawodów / treningów", która będzie używana „na gorąco" podczas imprez sportowych. Wszelkie opcje konfiguracyjne i inne nie potrzebne do bierzącego użycia powinny zostać umieszczone w menu.

2. Statystyki obliczane dla zawodnika

   (a) HP Aplikacja powinna obliczać statystyki dla każdego saneczkarza z osobna, zarówno w odniesieniu do konkretnych zawodów, treningu jak również globalnie z uwzględnieniem wyników we wszystkich eventach w których brał on udział, która to statystyka ma być nazwana statystyką kariery.

   (b) HP Część parametrów statystycznych ma być dostępna tylko w statystykach per zawody (trening), część w statystykach ogólnych (statystykach kariery) a część zarówno w jednych jak i w drugich.

   (c) HP Jeżeli dany zawodnik startuje w jedynkach oraz jednocześnie w dwójce, to statystyki winny być obliczane osobno dla dwójki jako całość a osobno dla startu „pojedycznego".

   (d) HP Program winien wyszukiwać najsłabszy, najlepszy i średni czas ślizgu dla każdego zawodnika (dwójki) per konkurencja

   (e) HP Program powinien wyliczać średnią prędkość ślizgu na podstawie czasu śligu, oraz „Track Tune" czyli parametrów toru zawierącego m.in.

3. Obsługiwane konkurencje saneczkowe i ich specyficzne wymagania

4. Parametry toru (Track Tunes)

5. Funkcjonalności Ekranu Głównego

6. Funkcjonalności modułu „Menadżer Zawodników"

7. Funkcjonalności modułu „Obsługa Zawodów / Treningów"

8. Format zapisu danych

9. Zabezpieczenia przed utratą danych

Rozdział 4

Terminarz (Timing Plan)

„Kto sieje ASAP ten zbiera FUCKUP”.

„Yyyyyy.... Actually... yyyyy... No progress”.

„Do some magic”.

„Poland one step back”

Typowe rozmowy programistów kiedy do terminu wydania pozostało kilka dni, nic nie działa i nikt nie wie dlaczego.

Rozdział 5

Ekran Startowy

Niniejszy rozdział jest pierwszym, który tak naprawdę traktuje o samej obsłudzie programu MKS_JG. W każdy rozdziale zostanie szczegółowo omówiony ekran, którego rozdział dotyczy. Opisane zostaną funkcję poszczególnych przycisków, list rozwijanych, checkboxów, menu itp itd. Zamieszczę też przykładowe scenariusze opisujące krok po kroku jak wykonać konkretną akcję. Na początek przyjrzyjmy się ekranowi startowemu.

Rysunek 5.1: Ekran startowy z naniesionymi etykietami

Służy on przede wszystkim do uruchamiania podstawowych modułów funkcjonalnych aplikacji MKS_JG, oraz ustawienia podstawowych opcji konfiguracyjnych, które muszą być określone jeszcze przed uruchomieniem jakichkolwiek modułów. Wynika to z faktu, że moduły te są inicjalizowane w konkretny sposób w zależności od ich (tj. tych ustawień) nastawienia.

5.1 Dostępne opcje

Ekran startowy ma w swoich założeniach być prostym elementem służącym do uruchamiania poszczególnych modułów funkcjonalnych. Znajdują się na nim nazwa programu oraz numer wersji wraz z datą kompilacji. Jest też podany mój nr telefonu komórkowy, co ma umożliwić użytkownikowi bardzo szybkie uzyskanie pomocy w sprawie używania MKS_JG. Pozostałe opcje oznaczone etykietami to:

1. Pasek menu. Warto pamiętać, że będzie on obecny w tym miejscu tylko na systemach Windows i Linux. Na komputerach Apple z systemem operacyjnym MacOS X pasek menu będzie widoczny tam gdzie zawsze, czyli na belce górnej poza samym oknem aplikacji. Dokładne omówienie pozycji menu zamieszcze w kolejnych sekcjach.

2. „MENADŻER ZAWODNIKÓW". Przycisk uruchamiający Menadżera Zawodników. Jest to moduł służący do zarządzania zawodnikami. Oprócz typowych opcji dodawania, usuwania oraz modyfikacji ich danych są (będą) w nim dostępne obszerne statystyki obliczane na podstawie wszystkich zawodów i treningów zapisanych w plikach danych. Więcej informacji na temat obsługi Menadżera Zawodników znajduje się w stosownym mu rozdziale

3. „OBSŁUGA ZAWODÓW I TRENINGÓW". Przycisk uruchamiający moduł służący do obsługi zawodów i treningów. Jest on głównym elementem MKS_JG umożliwiającym definiowanie różnego typu konkurencji rozgrywanych w ramach zawodów czy treningów. Posiada funkcjonalność rejestracji czasu ślizgu dla każdego saneczkarza czy to przez ręczne wprowadzanie, czy to przez współprace z zewnętrznym chronometrem. Generuje raporty zarówno w formacie CSV dla eksportu do arkusza kalkulacyjnego jak i „oficjalne" w formacie PDF, które mogą być z powodzeniem publikowane.

4. Pasek statusów. Wyświetla obecnie ustawiony język interfejsu oraz tryb pracy aplikacji.

Ekran startowy jest to główny element MKS_JG, który jest tworzony przez metodę *static void main();* będącą punktem wejścia do apikacji. Przekładając to na język zrozumiały dla każdego oznacza to, że ***zamknięcie Ekranu Startowego spowoduje wyłączenie całej aplikacji MKS_JG***

5.2 Tryby pracy aplikacji

MKS_JG posiada dwa główne tryby pracy odnoszące się do sposobów w jaki przechowywane są wszystkie dane na których pracuja ta aplikacja. Chodzi tutaj o dane saneczkarzy, konfigurację poszczególnych treningów i zawodów, oraz przeprowadzanych w nich konkurencji. Dostępne są dwa:

- Zapis wszystkich danych w bazie danych SQL
- Zapis wszystkich danych w plikach XML (jeden dla zawodników + n dla zawodów/treningów)

Każdy z tych trybów ma swoje wady i zalety. Choć dane są dostępne w obydwu przypadkach w taki sam sposób i poza pewnymi szczegółami użytkownik nie widzi absolutnie żadnych różnic w obsłudze MKS_JG, to są to zupełnie różne i odmienne koncepcje.

Baza danych SQL - Dane zawodników, statystyki, treningi itp itd. są zapisywane w relacyjnej bazie danych SQL. Nie zaciemniając tutaj nadmiarem informacji technicznych wystarczy powiedzieć, że baza danych SQL jest to dedykowana aplikacja służąca do przechowywania danych numerycznych, tekstowych czy nawet graficznych albo binarnych w zorganizowanych strukturach

danych nazywanych tabelami. Relacyjność polega na tym, że dane zapisane w tabelach można ze sobą skojarzyć, tj. połączyć relacją. Jeżeli w jednej tabeli znajdują się dane saneczkarzy (Imię, Nazwisko itp) to w tabeli konkurencji znajduje się tylko referencja na konkretnych zawodników w niej startujących. Przy „wyciąganiu" danych z bazy zapytanie kojarzy dane z kilku różnych tabel i tworzy jeden spójny wynik. Dzięki temu przy np. konieczności aktualizacji informacji zmianę dokonuję się w jednym a nie np w pięciu różnych miejscach, bo pozwala zachować spójność. *W praktycznym ujęciu zaletą bazy SQL jest możliwość pracy na tych samych danych na różnych komputerach, również w tym samym czasie.*

Pliki XML - Wszystkie dane programu są zapisywane w plikach tekstowych XML (Extensible Markup Language). Plik XML przypomina bardzo HTML czyli język opisu stron internetowych, jednakże może on zawierać dowolne dane. W przypadku MKS_JG używane jest nie jeden ale wiele plików XML (conajmniej dwa). Jeden plik XML jest używany do przechowywania danych zawodników i ich statystyk. Kolejne pliki XML są używane do przechowywania informacji na temat zawodów / treningów i rozgrywanych w ich ramach konkurencji. Obowiązuje tu zasada, że jeden trening lub zawody to jeden plik XML. Między plikami istnieją pewne relacje, np. plik zawodników przechowywuje wprawdzie łączną liczbę ślizgów saneczkarza podzieloną na poszczególne tory i bramki startowe, ale nie przechowuje szczegółów dotyczących poszczególnych treningów i zawodów. Wskazuje jedynie w pliku o jakiej nazwie i wewnętrznym identyfikatorze znajdują się potrzebne wyniki.

5.3 Wady i zalety użycia bazy danych SQL

Zalety:

- Ponieważ baza danych jest osobną aplikacją udostępniająca przechowywane przez siebie dane przez specjalny interfejs wykorzystujący TCP/IP, umożliwia ona prace kilku instancji MKS_JG na kilku różnych komputerach w tym samym czasie.
- Baza danych może być zlokalizowana na dowolnym, odległym serwerze, niekoniecznie na tym samym komputerze na którym używa się MKS_JG.
- Baza danych umożliwia uniknięcie problemu przenoszenia danych pomiędzy komputerami i problemów z zachowaniem ich spójności oraz wspólnej wersji na wszystkich maszynach.
- Dane przechowywane w bazie danych są lepiej zabezpieczone przed uszkodzeniem niż pliki XML.

Wady:

- Baza danych SQL dowolnego typu (MySQL, PostgreSQL itp) jest osobną aplikacją, którą trzeba gdzieś zainstalować. Albo na tym samym komputerze, który będzie używany do pracy z MKS_JG albo na osobnym serwerze.
- Jeżeli baza danych będzie zainstalowana na odległym serwerze do którego aplikacja będzie łączyła się przez Internet, to podczas całej pracy MKS_JG komputer będzie musiał posiadać stałe połączenie z Internetem. Zerwanie połączenia wprawdzie nie uszkodzi już zapisanych danych ale uniemożliwi dalszą pracę MKS_JG.
- Wprawdzie baza danych SQL używana w MKS_JG nie powinna przechowywać bardzo dużej ilości danych, to jednak sam silnik bazodanowy będzie zapewnie używał większej ilości zasobów (głównie pamięć operacyjna), niż MKS_JG ładujący wszystkie dane z plików XML do pamięci.

- Serwer bazy danych SQL jest z natury aplikacją, która działa non stop. Będzie się więc standardowo uruchamiał automatycznie wraz z systemem operacyjnym nawet jeżeli MKS_JG nie będzie używany. Ręczne zatrzymywanie i uruchamianie serwera bazy danych może być nieco uporczywe.

5.4 Wady i zalety użycia plików XML

Zalety:

- Pliki XML naturalnie nie wymagają instalacji żadnego zewnętrznego oprogramowania.
- Pliki XML można w łatwy sposób archiwizować i tworzyć ich kopie zapasowe. Wystarczy użyć np. archiwizera WinZIP albo WinRAR. Odtworzenie kopii zapasowej wymaga po prostu podmienienia odpowiednich plików.
- W tej konfiguracji MKS_JG może być „trzymany" na pendrive razem z plikami XML. Umożliwia to łatwe przenoszenie pomiędzy różnymi komputerami. UWAGA! Ze względu na względnie dużą awaryjność pamięci USB (i możliwość ich zgubienia) zaleca się oczywiście wykonywanie kopii zapasowych na innym nośniku.
- Ponieważ MKS_JG nie będzie ładował wszystkich danych z wszystkich plików XML do pamięci zużycie zasobów będzie mniejsze niż dla bazy danych.

Wady:

- Praca na kilku komputerach na raz jest niemożliwa, nawet jeżeli pliki XML będą umieszczone na współdzielonym zasobie. Próba siłowej pracy kilku instancji MKS_JG na tych samych plikach może szybko doprowadzić do uszkodzenia danych.
- Pomimo tego, że plik XML ma poniekąd postać zrozumiałą człowieka (w odróżnieniu od nieczytelnych plików biarnych bazy danych) to nie wolno go ręcznie edytować, jeżeli nie ma się absolutnie pewności co się robi. Ręczne modyfikację plików XML mogą spowodowac ich uszodzenie.
- Pliki XML wymuszają większą samodyscyplinę użytkownika przy przenoszeniu danych pomiędzy komputerami. Pojawia się bowiem oczywisty problem wersjonowania tychże plików, który bez właściwej troski może doprowadzić do powstania kilku różnych plików XML z różnymi danymi. Ich połączenie w jeden aktualny zbiór może być bardzo utrudnione, a na pewno będze wymagało ręcznych modyfikacji.
- Ponieważ całość danych podzielona jest na kilka plików XML to wszystkie one muszą być trzymane w całości w jednym miejscem. Zgubienie np. jednego z plików z wynikami treningów spowoduje utratę części danych w statystykach saneczkarzy.

Rozdział 6

Menadżer Zawodników

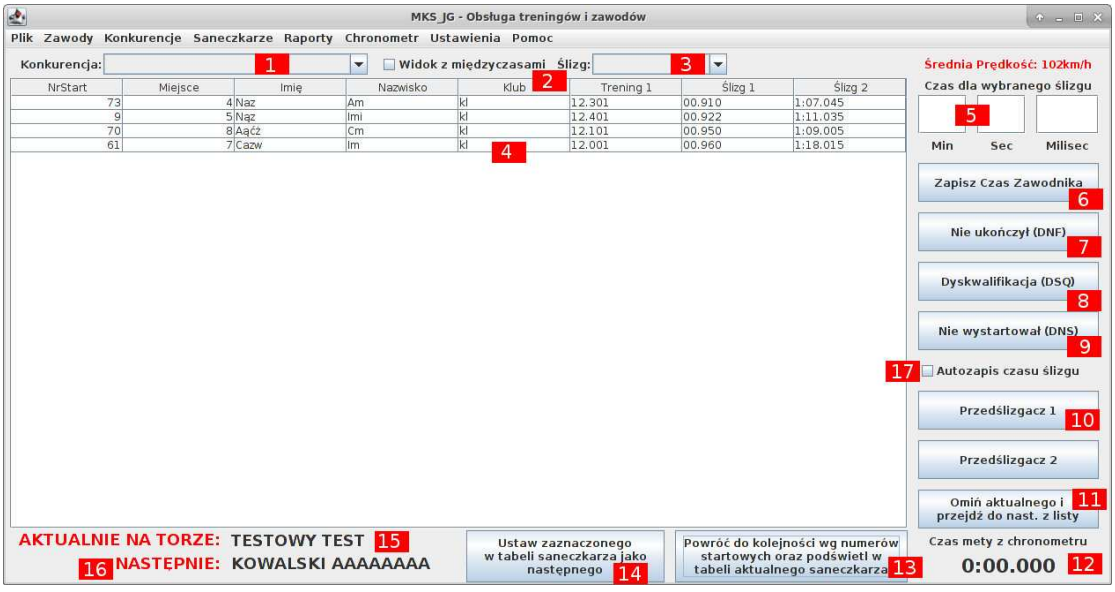
// TODO

Rozdział 7

Obsługa Treningów i Zawodów

„Paralotniarze dzielą się na tych którzy lądowali na drzewie i na tych, którzy dopiero będą” - popularne powiedzenie paralotniowe. Aby rzucić nieco światła na sens tego porzekadła należy wyjaśnić, że paralotniarze są dość oryginalną kategorią lotnictwa. O ile w przypadku szybowców, samolotów a nawet lotni zakończenie lotu w koronach drzew uznawane jest conajmniej za poważny incydent jeżeli nie jako wypadek lotniczy, to u paralotniarzy jest to po prostu „inna technika lądowania”. Oczywiście nie jest to nic porządanego i lądowanie na drzewie nigdy nie jest traktowane jako norma. Z drugiej jednak strony często drzewa okazują się bezcennym ratunkiem dla paralotniarza, chroniąc przed zderzeniem z ziemią po masywnym podwinięciu paralotni na małej wysokości.

Obsługa Treningów i Zawodów to w zasadzie główny moduł aplikacji MKS_JG, który będzie zapewnie wykorzystywany w największym stopniu. Wygląd jego okna głównego przedstawia się na rysunku 7.1 7.2 i 7.3



Rysunek 7.1: Widok ekrany głównego Obsługi treningów i zawodów

Rysunek 7.2: Zbliżenie na przyciski po prawej stronie okna

Rysunek 7.3: Zbliżenie na panel w dolnej części okna

7.1 Dostępne opcje

Wygląd i dostępne opcję w oknie obsługi zawodów i treningów w programie MKS_JG zależą nieco od konkretnych opcji konfiguracji które wybierze użytkownik, oraz rodzaju rozgrywanej konkurencji (dwójki / jedynki). Widoki znajdujące się na rysunkach 7.1 7.2 i 7.3 przedstawiają najbardziej ogólny zarys interfejsu. Różnice, które mogą się pojawić będę opisywał w stosownych punktach z wyjaśnieniem kiedy może zajść takowa różnica.

1. **Menu wyboru konkurencji** - Jeżeli na zawodach albo na treningu rozgrywa się więcej niż jedną konkurencję (np. dwójki, juniorzy, juniorki itp), to wspomniane pole rozwijane służy do przełączania się pomiędzy nimi. Wybór którejkolwiek pozycji z listy automatycznie przeładowywuje główną liste wyników, powodując wczytanie odpowiedniej (ówcześnie wprowadzonej) listy startowej. Kursor w tabeli oraz informacja „Aktualnie na Torze" jest ustawiana na pierwszego saneczkarza który nie ma zarejestrowanego czasu ślizgu (na pierwszego który jeszcze nie wystartował)

2. **Widok z międzyczasami** - Opcja ta przełącza widok listy startowej / wyników na wariant z międzyczasami. Po jej zaznaczeniu na liście nie pojawiają się czasy wszystkich ślizgów punktowanych i niepunktowanych ale jedynie dane o ślizgu wybranym z pola rozwijanego oznaczonego numerem 3. Za kolumną „klub" pojawiają się kolejne międzyczasy, których ilość jest zależna od ustawień chronometru (maksymalnie 3 punkty pomiaru oprócz czasu na mecie). Przedostanią kolumną jest łączny czas ślizgu zmierzony na linii mety. Ostatnią kolumną jest zawsze albo prędkość średnia albo prędkość chwilowa, rozumiana jako prędkość obliczona pomiędzy dwoma punktami międzyczasu.

3. **Ślizg** - Menu służy do przełączania się pomiędzy wieloma zdefiniowanymi ślizgami. W przypadku widoku z międzyczasami przeładowywuje cały widok listy wyników. Bez użycia tej opcji ustawia kursor automatycznie na pierwszego zawodnika w wybranym z listy ślizgu (pierwszego wg listy startowej, lub pierwszego który jeszcze nie startował i nie ma zanotowanego czasu). Analogicznie używanie przycisków strzałek na klawiaturze, albo klikanie myszą na kolumny odpowiadające kolejnym ślizgom zmienia wartość wyświetlaną w tym polu rozwijanym - kliknięcie jakiegokolwiek czasu w kolumnie „Treningowy 2" powoduje ustawienie tego pola rozwijanego na „Treningowy 2".

4. **Główna lista z wynikami** - Główny element interfejsu Obsługi zawodów i treningów, w którym wyświetlane są tabelaryczne informację na temat tego, co dzieje się aktualnie na torze sankowym. Oprócz imienia i nazwiska oraz przynależności klubowej pojawiaja się tam oczywiście wyniki pomiaru czasu. W zależności od zaznaczenia lub nie opcji „Widok z Międzyczasami" pojawią się tam albo czasy mety dla wszystkich ślizgów treningowych i punktowanych, albo poszczególne czasy pośrednie (start, pierwszy, drugi, meta) dla ślizgu ustawionego z menu „Ślizg" (patrz pkt 3). ***Lista jest w pełni sortowalna po dowolnie wybranej kolumne. Domyślnie prezentowane dane są pozostowane wg numerów startowych rosnąco, aczkolwiek użytkownik może ten aspekt dowolnie zmieniać.*** Wystarczy kliknąć na nagłówek wybranej kolumny aby przełączać się między kolejnością rosnącą a malejąca. Zasada ta tyczy się zarówno kolumn przechowywujących dane numeryczne (czas ślizgu dla saneczkarza), jak również dane tekstowe (imię, nazwisko itp). Informacje są oczywiście traktowane jako jedna całość, dlatego kliknięcie na dowolny nagłówek powoduje posortowanie wszystkich danych.

5. **Czas wybranego ślizgu** - Zestaw trzech pól służących do edycji i zapisywania wybranego czasu ślizgu, osobno dla minut, sekund oraz milisekund. Działa ono w dwojaki sposób w

zależności od tego czy MKS_JG jest skonfigurowany do używania chronometru, czy też nie. Jeżeli praca z chronometrem jest wyłączona to w polach tych będzie ukazywał się po prostu czas wybranego z listy 4 ślizgu. Tzn. jeżeli użytkownik kliknie w tableli na przykład czas pierwszego ślizgu treningowego dla zawodnika Adam Kowalski, to zostanie on w tym miejscu wyświetlony i będzie mógł być z niego edytowany. Jeżeli obsługa pomiaru czasu zostanie włączona to oprócz wymienionej już funkcjonalności, czyli edycji wskazanego w tabeli czasu będą tam pojawiały się automatycznie czasy obliczone na podstawie czasu z chronometru. **W momencie gdy saneczkarz przetnie linie mety i chronometr wyśle o tym inormację do aplikacji MKS_JG, kursor w tabeli 4 zostanie automatycznie przestawiony na ślizg i zawodnika który właśnie ukończył bieg. Pola „Czas wybranego ślizgu" zostaną zaktualizowane automatycznie o dane obliczone na podstawie chronometru.** Czas zostanie zapisany na liście i w danych programu po akceptacji przez użytkownika, lub automatycznie po 5 sekundach jeżeli zaznaczona jest opcja 17 „Autozapis czasu ślizgu". Jeżeli w tabeli 4 zostaną wybrane dane nienumeryczne, tj. np. Imię bądź Nazwisko saneczkarza to pole te zostaną wyczyszczone.

6. **Zapisz czas zawodnika** - Kliknięcie tej opcji potwierdza zapis do tabeli aktualnie modyfikowanego czasu ślizgu. Warto nadmienić, że kliknięcie podczas edycji na jakikolwiek inny czas lub otrzymanie danych z chronometru powoduje automatyczne odrzucenie edytowanych zmian i albo przełączenie na wskazany kursorem czas ślizgu, albo przełączenie na ślizg i zawodnika który właśnie ukończył swój start. W drugim przypadku kliknięcie przycisku służy do potwierdzenia, że otrzymany z chronometru czas przejazdu jest poprawny i może zostać temu saeczkarzowi zapisany.

7. **Nie ukończył (DNF)** - Powoduje zapisanie aktualnie wybranemu z tabeli 4 zawodnikowi DNF jako wynik ślizgu. Warto pamiętać, że użycie DNF **nie** wymaga potwierdzenia przy użyciu „Zapisz czas zawodnika"

8. **Dyskwalifikacja (DSQ)** - Powoduje zdyskwalifikowanie zawodnika z konkurencji i zawodów. Po użyciu tej opcji zawodnik będzie pomijany na liście startowej a wszystkie jego ślizgi zostaną oznaczone jako DSQ.

9. **Nie wystartował (DNS)** - Analogicznie jak w punkcie 7 ale powoduje zapisanie DNS dla ślizgu.

10. **Przedślizgacz 1** - Przycisk służący do ustawienia pierwszego przedślizgacza jako aktualnego, lub następnego startującego. Po przypisaniu osób do pełnienia roli pierwszego i drugiego przedślizgacza opis przycisku zostanie zaktualizowany do formy: *„Przedślizgacz Imię Nazwisko"*. Jeżeli program otrzymał z chronometru informację o przecięciu startu przez zawodnika (tor jest zajęty) to przedślizgacz zostanie ustawiony jako następny startujący („Natępnie"). W przeciwnym razie, tj. gdy tor jest wolny lub MKS_JG ma wyłączoną obsługę chronometru przedślizgacz zostanie ustawiony jako „Aktualnie na Torze". **Przedślizgacz jako osoba nie startująca w konkurencji nie widnieje na liście i nie jest uwzględniany w głównym raporcie wyników, jednakże MKS_JG normalnie rejestruje jego czas ślizgu. Czas ten jest doliczany do statystyk saneczkarza istnieje też możliwość wygenerowania pomocniczego raportu z wynikami obudwy przedślizgaczy.**

11. **Przedślizgacz 2** - Analogicznie jak dla pierwszego przedślizgacza.

12. **Omiń aktualnego i przejdź do następnego z listy (startowej)** - Użycie opcji powoduje przeskoczenie do kolejnego saneczkarza na liście startowej. Jeżeli MKS_JG ma

włączoną obsługę chronometru, to naciśnięcie tego przycisku zresetuje wewnętrzną maszynę stanu pomiaru czasu. Oznacza to, że jeżeli aktualnie na torze jedzie jakiś zawodnik, to MKS_JG przechodząc do kolejnego z nich całkowicie zignoruje fakt przecięcia linii mety. Jeżeli użycie opcji nastąpiło omyłkowo, czas może zostać dopisany ręcznie przez kliknięcie na ślizgu ominiętego zawodnika, wprowadzenie czasu (minuty, sekundy, milisekundy) i kliknięcie „Zapisz czas dla zawodnika". Jeżeli przy wielokrotnym użyciu tej funkcji kursor dojdzie do ostatniego zawodnika w danym ślizgu, to kolejne jego naciśnięcie wróci do najniższego numerem startowym, który nie ma zapisanego czasu ślizgu. Funkcja z definicji będzie omijała saneczkarzy, którzy mają zapisany czas.

13. **Powróć do kolejności wg numerów startowych oraz podświetl w tabeli aktualnego saneczkarza** - Naciśnięcie tego przycisku skutkuje powrotem MKS_JG do kolejnego zawodnika wg. listy startowej, czyli najniższego numerem startowym saneczkarza który w danym ślizgu nie ma zapisanego czasu. Obowiązuje tu podobna zasada przełączania co w przpadku przedślizgaczy, tj. jeżeli MKS_JG pracuje z chronometrem i otrzyma od niego informację o starcie to „zawodnik powrotny" będzie ustawiony jako następny. W przeciwnym razie zostanie zamieniony z „Aktualnie na Torze".

14. **Ustaw zaznaczonego saneczkarza jako następnego** - Funkcja pozwala na modyfikowanie „ad-hoc" kolejności startowej zawodników jeżeli wymaga to aktualna sytuacja. Jej wybranie powoduje ustawienie aktualnie podświetlonego w tabeli wyników zawodnika jako następnego w kolejce startowej. Należy pamiętać, że kliknięty w tabeli saneczkarz jest ZAWSZE ustawiany jako następny bez względu na to czy program współpracuje z chronometrem i czy tor jest wolny albo zajęty. Jeżeli sprzętowy pomiar czasu nie jest sprzęgnięty z MKS_JG, to po naciśnięciu tego przycisku należy ewentualnie użyć opcji 12 (Omiń aktualnego i przejdź...), która w tym konkretnym przypadku nie przejdzie do następnego saneczkarza wg. numerów startowych ale do osoby wskazanej przez aktualnie omawianą funkcję.

15. **Aktualnie na Torze** - Imię i Nazwisko saneczkarza który w tym momencie znajduje się na starcie (przed linią startu) albo aktualnie jedzie. W przypadku konkurencji dwójkowych pojawiają się tu jedynie nazwiska w kolejności góra - dół.

16. **Następnie** - Analogicznie jak punkt 15.

17. **Autozapis Czasu Ślizgu** - Włącza automatyczny zapis czasów otrzymanych z chronometru. Po przecięciu przez zawodnika linii mety

kjkj

Rozdział 8

Informcje kontaktowe

// TODO

Rozdział 9

Posłowie

//TODO

9.1 Inżynieria Oprogramowania jako metoda dorobku dla Saneczkarza

Na sam koniec instrukcji obsługi chciałbym przybliżyć nieco arkana pracy jako Inżynier Oprogramowania (Programista) i opisać jak ten zawód może pomóc zabezpieczyć finansowo karierę sportową typowego saneczkarza. Do tegoż wywodu skłoniły mnie liczne wywiady i konferencję prasowe, na których w zasadzie w każdym przypadku członkowie kadry seniorskiej wspominali o mniejszych bądź większych problemach finansowych, jak również oczywistej wątpliwości: „Co dalej po sankach?". Nie mam żadnej większej wiedzy o saneczkarstwie i o tym jak wygladają kulisy funkcjonowania i finansowania kadry, wydaje mi się jednak że przy pewnych założeniach nauka programowania może się typowemu zawodnikowi bardzo przydać.

Na początek opiszę po krótce jak to wyglądało w moim przypadku. Z racji tego, że mój śp. Ojciec był z zawodu elektrykiem i w zasadzie zawsze pracował w okolicach tej branży, to od dziecka miałem już wyrobione przekonanie o tym co chcę robić w swoim życiu. Miałem pewność, że musi to być cokolwiek związanego albo z prądem elektrycznym w dowolnej postaci albo konkretnie z komputerami. Po liceum ogólnokształcącym (klasa mat-fiz) poszedłem oczywiście na studia, dokładnie na Politechnikę Rzeszowską. Po pierwszym stopniu jestem inżynierem elektrykiem. Po drugiem stopniu magistrem inżynierem elektroniki i telekomunikacji. Brzmi strasznie ale nie było tak źle jak to się może wydawać. *Najważniejsze jest jednak to, że podczas studiów de facto nie byłem uczony stricto programowania. Oznacza to, że brak studiów Informatycznych nie przekreśla szansy na zostanie programistą.* Jedyną styczność z tworzeniem oprogramowania miałem podczas zajęć z Informatyki na pierwszym roku, oraz z techniki mikroprocesorowej na trzecim. Po tych dwóch przedmiotach byłem już prawie pewien, że chcę zostać programistą. Żeby jednak rozwiać wątpliwości. *Ja nigdy nie byłem specjalnym orłem z przedmiotów ścisłych.* W liceum na matematyce jechałem na trójach, a gdy moja nauczycielka tego przedmiotu (i jednocześnie wychowawczyni) usłyszała że chcę zdawać podstawową maturę z matematyki (byłem ostatnim rocznikiem dla którego ten przedmiot nie był obowiązkowy) to wprost wyrażała obawy, że ja tego egzaminu nie zdam. A nawet jeżeli zdam i dostanę się na politechnikę to na pewno sobie nie dam rady. *Piszę to aby pokazać, że nie trzeba być niewiaomo kim aby zostać programistą.*

Jak więc nie-programista i z formalnego punktu widzenia nie-informatyk może się przekwalifikować do nowego zawodu? Po pierwsze należy określić sobie ogólne cele i rozpoznać lokalny

rynek pracy, co zaważy na technologiach i jezykach, których powinno się uczyć. Potencjalny programista-saneczkarz z Dolnego Śląska jest we względnie komfortowej sytuacji. W samej Jeleniej Górze jest w zasadzie jedna jedyna firma programistyczna. Nazywa się CodeTwo i zajmuje się tworzeniem oprogramowania dla Microsoft Office 365 i Microsoft Exchange (oprogramowanie serwera pocztowego). Naturalnie więc używa języka C# i ogólnie technologii oraz bibliotek firmy Microsoft. Sugerowało by to, że jeżeli ktoś chciałby zdecydowanie pozostać w JG, to powinien docelowo przejść w stronę MS aby móc doprowadzić się do poziomu w którym realne stanie się dostanie pracy jako Junior.

Stolicą branży programistycznej na Dolnym Śląsku jest jednak Wrocław. We Wrocławiu firm jest „do wyboru do koloru" w każdej używanej obecnie technologii, zaczynając od ogólnych „technologii webowych" czyli PHP, JavaScript, nodeJS idąc przez C++/Java/C# a kończąc na sterownikach mikroprocesorowych i generalnie branży zachaczającej znacznie o elektronikę. Są to zarówno potężne korporacje takie jak Nokia Networks, Ericsson czy Hewlett-Packard Enterprise, jak również małe tzw. „software hourse", czyli firmy zatrudniające do kilkunastu pracowników i obsługujące zdecydowanie mniejsze zlecenia.

Jednak jak wygladają konkrety? ***Co ma dać saneczkarzowi zostanie programistą i przede wszystkim czy i jak da się to pogodzić z karierą sportową?*** Jak wspomniałem nie znam szczegółów życia i pracy zawodowego saneczkarza, aby określić konkrety musiałbym porozmawiać z zainteresowanym. ***Jednakże generalnie uważam, że pogodzenie tych dwóch rzeczy jest jak najbardziej możliwe.*** Zalety tej branży sa następujące:

- **Pieniądze** - Nie ma co ukrywać, że obecnie w Polsce, Europie i na Świecie branża IT jest jedną z najlepiej opłacanych. W samej branży IT najlepiej zarabiają właśnie programiści. Na przykładzie własnego życia mogę potwierdzić, że zmiana pracy, przeprowadzka do Bielska-Białej i generalnie wejście typowo w rolę Inżyniera Oprogramowania otworzyło przede mną możliwości, których wcześniej nie miałem. Pracując w Rzeszowie w roli wdrożeniowca już sam przyjazd samochodem w Karkonosze był sporym wypadkiem (530km w jedną stronę). **Zaznaczam jednak aby nie ulegać „wykopowym" opowieściom o mitycznych 10 000 złotych miesięcznie na rękę, które mają zarabiać młodzi programiści bez żadnego wysiłku. To nie jest prawda a wierutne kłamstwo!** Mi samemu brakuje do tej kwoty bardzo dużo i już samo znalezienie się w miejscu w którym jestem teraz kosztowało dużo pracy. Jeżeli zaś chodzi o same zarobki to zależą one od miasta i doświadczenia. We Wrocławiu, Katowicach, Krakowie, Warszawie gdzie koszta życia są dużo większe zarobki również będą naturalnie większe. Poza dużymi miastami są nieco mniejsze. Podobnież jak w sporcie tu również są pewne „stopnie".

  - Zaczyna się często od stażysty czyli takiego młodzika. Stażysta to osoba, która ma bardzo małą wiedzę i którą w zasadzie trzeba cały czas uczyć i poświęcać dużo czasu. Stażysta z definicji przychodzi do firmy na dość krótki czas, często robi za darmo albo za dość małe kwoty 1000~1300pln netto miesięcznie.
  - Kolejnym etapem rozwoju jest Junior. Junior to osoba, która zna teoretycznie język i technologię którą się posługuje. Ma jakieś bardzo małe doświadczenie we własnych, małych projektach tworzonych często tylko w celach ćwiczebnych (czyli projekty, które nigdy nie weszły do produkcyjnego użycia) ale albo nie ma w ogóle doświadczenia zaowodowego, albo do doświadcze jest bardzo małe np. mniej niż rok. W pracy wykazuje się już dość dużą samodzielnością ale cały czas wymaga nadzoru kogoś drugiego, kto będzie w stanie wychwytywać ewentualne błędy i będzie mu pomagać gdy nie będzie czegoś wiedział albo potrafił. Junior w zależności od technologii i miasta zarabia około 2000~3500 netto miesięcznie, choć w niektórych technologiach (Java) zdarzają się pojedyncze oferty na sumy nawet 5000 netto w Krakowie czy Wrocławiu.

  - Po około roku, półtorej (rzadko więcej) roku pracy na stanowisku Juniora przeskakuje się na poziom tzw. „Regulara" albo „Middle". Regular jest osobą, która jest już w pełni samodziela w swojej pracy. Zna bardzo dobrze język i technologię ale przede wszystkim, co jest jeszcze ważniejsze wie jak odnajdywać rozwiązania problemów z którymi się zmaga. Wie gdzie i jak szukać dokumentacji technicznej i wsparcia technicznego. Ma dużą wiedze na temat projektu nad którym pracuje i może bez problemu nawiązać współpracę z innymi ludźmi i zespołami nad nim pracującymi. Zarobi regulara zaczynają się od kwoty 3500pln netto dla małych miast (Rzeszów) a kończą się gdzieś w okolicach właśnie tych mitycznych 10 000pln netto miesięcznie.
  - Ostatnim etapem w karierze programisty jest Senior. Senior jest osobą, która ma lata (najczęściej 4 i więcej) doświadczenia w technologii którą używa. Wiedza i doświadczenie seniora jest tak duże, że może w zasadzie pełnić rolę kierownika projektu albo kierownika zespołu. W drobnych szczegółach zna projekt nad którym pracuje i jest ostatnią deską ratunku innych gdy „nic nie działa i nikt nie wie dlaczego". W niektórych technologiach zarobki Seniorów dochodzą do 15 000 złotych netto.

- **Elastyczny czas pracy** - Jest to benefit pozapłacowy, który jest pożądany przez pracowników w największym stopniu. Siedziby dużych firm IT najczęściej są ochraniane na miejscu przez ochroniarzy, oraz wyposażone są w zamki otwierane kartami zbliżeniowymi. Ten system jest często sprzęgnięty z pomiarem czasu pracy. Elastyczny czas pracy polega na narzuceniu jedynie ilości godzin, które pracownik musi przepracować w ustalonym okresie (najczęściej miesiąc). Ile godzin pracownik będzie pracował dziennie to już sprawa jego i ewentualnie kierownika zespołu. Znam też częste przypadki w odniesieniu do zawodników w sporcie paralotniowym, gdy umowa dżentelmeńska z kierownikiem umożliwiała pracę po 12 godzin w tygodniu, aby potem odebrać nadmiarowe godziny jako ekstra dni wolne nie zabierające urlopu wypoczynkowego (którego często już po prostu nie było ze względu na wyjazdy na zawody). Przenosząc do na realia sankowe -> Przy odpowiednim układzie z kierownikiem można by we wrześniu / październiku pracować dużo więcej w niektóre dni, aby potem mieć możliwość wyjazdów na treningi i Puchar Świata czy inne zawody mając tylko 26 dni robocze urlopu do dyspozycji.

- **Praca zdalna** - Praca zdalna polega na pracy nie w siedzibie pracodawcy ale zdalnie, w dowolnie obranym miejscu. Dla jednych programistów jest to jedyny wyznacznik przy szukaniu zatrudnienia, inni nigdy by się jej nie podjęli. Dla pracodawcy zaletą są przede wszystkim niskie koszta własne. Pracownikowi nie trzeba wynajnować biura, co oczywiście kosztuje dość znaczna sumę pieniędzy. Często pracownik pracuje na własnym sprzęcie co też zmniejsza koszta pracodawcy. Zdalny programista jest oczywiscie z definicji mniej wydajny, bo ma utrudniony kontakt z resztą zespołu ale jak to pewna osoba mawia „korporacja to przewidziała". Jak to przenieść na realia sankowe? W dość oczywisty sposób. Pracować można tak samo z Jeleniej Góry jak z Altenbergu, Oberhofu, Koenigsee, Iglis, czy dowolnego innego miejsca na świecie. Oczywiście są pewne ograniczenia zarówno ze strony programistycznej, jak i pewnie sankowej (skupienie na starcie itp itd) ale nie jest to niemożliwe.

Teraz nieco o tym czy jest trudno :) Z programowaniem jest podobnie jak np. z paralotniarstwem i zapewne tak jak z sankami - aby osiagnąć sukces trzeba dużo pracować. Każdy zaczyna od podstaw, a potem z biegiem czasu nabiera doświadczenia i wiedzy, które pozwalają mu coraz więcej uzyskać. Pewne cechy saneczkarza mogą się wbrew pozorom okazać przydatne w karierze programisty i sprawić, że nie będzie to takie trudne jak może się wydawać ;).

Wprawdzie Góra Parkowa w Krynicy to nie start męski w Siguldzie ale domyślam się (bo wiedzy w temacie nie mam), że przy prędkości przelotowej 70 węzłów nie ma zbytnio czasu na dogłębne przemyślenia „czy ja na ten wiraż wjadę bardziej środek-lewa czy środek-prawa". Wydaje się (mi), że w tym sporcie kluczem jest daleko posunięta wyobraźnia przestrzenna i zdolność abstrakcyjnego myślenia pozwalającego wyborazić sobie linię przejazdu przez wszystkie wiraże i wstępne przeprocesowanie tego co trzeba zrobić podczas jazdy. **Tak się składa, że te same cechy czynią dobrego programistę :)** Kod programu zapisany w dowolnym jezyku programowanie jest jedynie jakimś suchym i abstrakcyjnym zapisem algorytmu wg. którego program ma działać. Programista musi umieć wyobrazić sobie jak ten kod programu będzie przekładał się na działającą aplikację, co dane operacje znaczą, co robią funkcję i np. dlaczego mają takie a nie inne argumenty. Ta sama zasadzia działą w drugą stronę, programista musi umieć wyobrazić sobie jak należy opakować w algorytmu i struktury danych wizję swoją albo klienta.