

II RIADENIE

Obsah

1 Úvod.....	II-1
2 Zloženie tímu.....	II-2
3 Projektový plán.....	II-3
3.1 Hrubý plán na zimný semester.....	II-3
4 Rozdelenie úloh v tíme.....	II-4
4.1 Úlohy členov.....	II-4
4.2 Autorstvo jednotlivých kapitol.....	II-4
5 Komunikácia v tíme.....	II-6
6 Podporné nástroje pre prácu v tíme.....	II-7
6.1 dotProject.....	II-7
6.2 Popis SVN.....	II-8
Príloha A - Ponuka.....	A - 1
Príloha B – Zápisy zo stretnutí.....	B - 1
Príloha C - Programátorské konvencie.....	C - 1
Príloha D – Percentuálne rozdelenie bodov.....	D - 1
Príloha E - Preberacie protokoly.....	E - 1

1 Úvod

V každom projekte, na ktorého riešení spolupracuje viac ľudí ako jeden je neoddeliteľnou súčasťou projektu riadenie. Riadenie projektu, ktorý vytvárame v predmete Tvorba softvérového systému v tíme je opísané v nasledujúcom texte. Táto časť dokumentácie je rozdelená do viacerých častí. V prvej sú predstavení členovia tímu. Druhá časť opisuje ako bola naplánovaná práca počas zimného semestra. V tretej časti sú opísané úlohy členov v tíme a ich podiel práce na vytvorení dokumentácie. Tiež sú tu popísané formy komunikácie v tíme a používanie podporných prostriedkov pre riadenie(dotProject, SVN) Súčasťou tejto časti dokumentácie sú aj prílohy, ktoré sa týkajú riadenia.

2 Zloženie tímu

Náš tím (tím číslo 13 – Švábi) tvoria šiesti členovia. Tím bol zriadený pre potreby predmetu Tvorba softvérových systémov v tíme. Všetci členovia sú absolventmi bakalárskeho stupňa štúdia na Fakulte informatiky a informačných technológií (FIIT) v Bratislave v odbore Informatika. Momentálne sú členovia tímu študentami FIIT v inžinierskom stupni štúdiu v odbor Softvérové inžinierstvo.

Členovia tímu:

Bc. Matúš Svrček,

Bc. Alexander Šimko,

Bc. Michal Štekláč,

Bc. Miroslav Štolc,

Bc. Jaroslav Tešlár,

Bc. Ľubomír Varga

Bližšie informácie o jednotlivých členoch tímu sú v prílohe Ponuka.

3 Projektový plán

Plán projektu na zimný semester sme vytvorili vo štvrtom týždni zimného semestra podľa harmonogramu pre predmet Tvorba softvérových systémov v tíme. Sú v ňom zahrnuté také činnosti, ktorých splnenie sme si stanovili za ciele pre jednotlivé týždne semestra. Projektový plán sme sa snažili naplňať v plnom znení, aby nevznikali zbytočné nezrovnalosti a časový sklz v projekte.

3.1 Hrubý plán na zimný semester

Týždeň semestra	Naplánované činnosti a ciele
4.-5.	<ol style="list-style-type: none"> 1. nájsť pokročilé techniky a stratégie pre hry reversi a go 2. zistiť, ako bolo vyriešené reversi pre hraciu plochu rozmerov 6x6 3. zistiť, ako bolo vyriešené go pre hraciu plochu rozmerov 5x5 4. preštudovať vylepšené algoritmy prehľadávania MiniMax stromu: alfa beta usekávanie, NegaMax, NegaScout, MTD(f), PVS (principal variation search), selektívne vyhľadávanie, transpozičná tabuľka, heuristika založená na histórii, Multicut alpha-beta pruning, PV splitting, Tree splitting with work stealing, Young brother wait concept 5. zanalyzovať, či bude možné BOINC-om vyriešiť komunikačný mechanizmus, ktorý potrebujeme
6.-7.	<ol style="list-style-type: none"> 1. návrh riešenia 2. návrh modulov systému 3. rozhodnutie výberu algoritmu pre MiniMax strom hry 4. analýza spôsobov, ako rozdeliť generovanie MiniMax stromu na workunit-y 5. konkretizácia zdieľania zdrojov, architektúra klienta a servera 6. tvorba dokumentácie – vytvoriť šablónu, rozdeliť kapitoly a zdokumentovať špecifikáciu požiadaviek, analýzu problematiky a hrubý návrh systému
8.	<ol style="list-style-type: none"> 1. odovzdanie dokumentácie 2. vypracovanie posudku pre konkurenčný tím
9.	<ul style="list-style-type: none"> ● odovzdanie posudku analýzy, špecifikácie a návrhu konkurenčného tímu
10.	<ul style="list-style-type: none"> ● dopracovanie zistených nedostatkov a návrh prototypu vybraných častí systému
11.	<ul style="list-style-type: none"> ● implementácia prototypu
12.	<ol style="list-style-type: none"> 1. odovzdanie prototypu vybraných častí systému spolu s dokumentáciou 2. používateľská prezentácia prototypu 3. testovanie vytvorených prototypov v skúšobnej prevádzke

4 Rozdelenie úloh v tíme

V tejto kapitole sú popísané úlohy jednotlivých členov v tíme, ich role. Tiež sa tu nachádza ktoré kapitoly sú dielom jednotlivých členov.

4.1 Úlohy členov

Na začiatku práce na projekte sme každému členovi tímu prideliť jeho hlavnú úlohu v rámci tímu. Toto rozdelenie sa nachádza v Tabuľka 1.

Člen tímu	Úloha
Bc. Matúš Svrček	Manažér podporných činností
Bc. Alexander Šimko	Vedúci tímu
Bc. Michal Štekláč	Manažér dokumentácie
Bc. Miroslav Štolc	Manažér vývoja
Bc. Jaroslav Tešlár	Manažér plánovania
Bc. Ľubomír Varga	Manažér kvality

Tabuľka 1: Úlohy členov v tíme

Každý z členov dostával počas práce na projekte aj čiastkové úlohy, ktoré sú podrobne opísané v zápisoch zo stretnutí. Na stretnutí boli stanovené úlohy pre členov a ich splnenie sa na nasledujúcom stretnutí vyhodnotilo.

4.2 Autorstvo jednotlivých kapitol

I Dokumentácia

1. Úvod - Bc. Alexander Šimko
2. Špecifikácia - Bc. Michal Štekláč
3. BOINC - Bc. Ľubomír Varga
4. Hry - Bc. Jaroslav Tešlár
5. Teoretický základ pre riešenie hier - Bc. Alexander Šimko
6. Existujúce riešenia - Bc. Michal Štekláč
7. Možnosti ukladania stromu na disk - Bc. Matúš Svrček
8. Návrh systému - Bc. Miroslav Štolc
9. Prototyp
 - 9.1 Prototyp hry na jeden PC - Bc. Miroslav Štolc
 - 9.2 prototyp aplikácie pre BOINC - Bc. Ľubomír Varga

10. Zhodnotenie – Bc. Alexander Šimko

11. Literatúra, Bc. Alexander Šimko

Prílohy:

Slovník pojmov - Bc. Michal Štekláč

Notácia - Bc. Miroslav Štolc

II Riadenie

1. Úvod – Bc. Michal Štekláč

2. Zloženie tímu – Bc. Michal Štekláč

3. Projektový plán – Bc. Jaroslav Tešlár

4. Rozdelenie úloh v tíme – Bc. Michal Štekláč

5. Komunikácia v tíme – Bc. Alexander Šimko

6. Podporné nástroje pre prácu v tíme

6.1. dotProject – Bc. Ľubomír Varga

6.2. SVN – Bc. Matúš Svrček

Prílohy:

Ponuka – Tím

Zápisy zo stretnutí – Tím, záverečná úprava Bc. Jaroslav Tešlár

Programátorské konvencie – Bc. Miroslav Štolc

Percentuálne rozdelenie bodov - Tím

Preberacie protokoly - Bc. Michal Štekláč

5 Komunikácia v tíme

Komunikácia je veľmi dôležitá zložka práce v tíme. Bez nej by žiaden tím nemohol fungovať. Počas nášho spoločného fungovania sme používali rôzne formy komunikácie.

Pravidelné stretnutia

Pravidelné stretnutia v utorok ráno 7:00 v softvérovom štúdiu na D bloku v priestoroch FIIT STUBA v Bratislave za prítomnosti všetkých členov tímu a pedagogického vedúceho, ktorý bol v pozícii zákazníka. Z týchto stretnutí sme vytvárali zápisy, ktoré sú voľne dostupné k nahliadnutiu. Ich úlohou je zachytiť vykonané rozhodnutia, prideliť úlohy na ďalší týždeň a zhodnotiť prácu za uplynulé obdobie.

Email

Komunikácia prostredníctvom elektronickej pošty sa používala v prípade, ak bolo potrebné, aby sa všetci členovia tímu oboznámili s novými skutočnosťami alebo ak bolo potrebné aby sa všetci vyjadrili k otázke, ktorá nebola naliehavá.

Skype

Túto konkrétnu službu hlasovej komunikácie prostredníctvom počítačovej siete sme používali v prípade potreby vyriešenia naliehavých otázok. Doteraz sme ju použili iba v priebehu prvého týždňa na vykonanie niekoľkých rozhodnutí v súvislosti s ponukou. Odvtedy všetka práca beží na základe týždenných plánov, ktoré sa stanovujú na pravidelných stretnutiach.

Komunikácia v reálnom čase prostredníctvom textových správ(angl. instant messaging)

Tento typ komunikácie sa využil na drobné konzultácie medzi jednotlivými členmi tímu alebo na pripomenutie, že má niekto niečo urobiť.

dotProject

Tento nástroj na manažment projektu slúžil na pridelovanie úloh a zdieľanie všetkej dokumentácie, zápisov, správ o vykonaní úloh. Slúžil nám na komunikáciu prostredníctvom vymieňania si dokumentov.

Osobné neformálne stretnutia

Veľká časť úvah, brainstormingu a rozhodnutí sa uskutočnila na osobných stretnutiach na internátnej izbe(časť tímu býva spolu na jednej izbe) alebo pri spoločných obedoch. Uvoľnenejšie prostredie sa ukázalo ako výdatný zdroj nápadov.

6 Podporné nástroje pre prácu v tíme

6.1 dotProject

Webová aplikácia dotProject¹ je nástroj na projektový manažment. Je určený pre používanie projektovými manažérmi a ich podriadenými pracovníkmi. Jedna inštancia tejto aplikácie môže obsahovať viacero spoločností a viacero projektov. Existuje v rôznych jazykových mutáciách a je dostupná (od verzie 2) pod licenciou GPL, teda zadarmo. Tento nástroj je vhodný a často i nasadzovaný v malých firmách a tímoch.

V našom tíme pokryl nástroj všetky naše potreby. Funkcionalitu tohto výborného nástroja naplno, využijú až tímy a firmy, ktoré majú okrem ľudských zdrojov i iné zdroje. Napríklad možnosť sledovať vyťaženosť miestností bola pre nás príjemným prekvapením, avšak sme ju nemali ako využiť. V tíme sme si určili niekoľko rolí. Každá rola mala vymedzené povinnosti a práva, ktoré bolo možné za pomoci ACL (Access Control List) priradiť priamo používateľom systému. Koordinácia ľudských zdrojov je za pomoci tohto systému jednoduchá a šetrí mnoho problémov.

Po každom stretnutí vytvorí za to zodpovedný člen tímu zápis a pridá, upraví, prípadne preplánuje úlohy v aplikácii dP (dotProject) podľa dohodnutých záverov zo stretnutia. Takto členovia tímu hneď večer po stretnutívidia zoznam svojich úloh, ich termíny odovzdania a predpokladané rozsahy. K úlohám sa môžu vyjadriť, poprípade si ich v prípade nutnosti upraviť. Systém v prípade úprav je schopný rozoslať všetkým dotknutým členom tímu mail o danej zmene.

Úlohy nie sú však ani zďaleka jedinou podpornou funkciou, ktorú náš tím aktívne využíva. K úlohám je pri ich postupnom plnení možné pridávať zápisy o činnostiach na nich. Z týchto zápisov (logov) je potom možné takmer plne automaticky vygenerovať projektový denník. Takisto je možné z týchto informácií vyvodiť vyťaženie jednotlivých členov tímu, ich efektívnosť a podobne. Ďalšou intenzívne využívanou funkcionalitou v dP je možnosť k úlohám pridávať súbory. Súbory môžu byť viacerých typov. Napríklad dokumenty alebo aplikácie. Vzhľadom na to, že ako tím používame tímový repozitár na svn serveri, aplikačné súbory do dP neukladáme. V dP aplikácii však udržujeme dokumentáciu projektu, ktorá vzniká zväčša na základe priradených úloh a tak je možné vždy k úlohe vytvoriť časť dokumentácie priradiť i rozpracovaný a následne výsledný súbor. Pri práci viacerých členov tímu na jednom dokumente využívame možnosť „check out“ a „check in“. Príkazom chcek out po zadaní dôvodu zamkneme daný súbor pri úprave a následne cez chcek in, uverejníme novú, upravenú, verziu dokumentu.

Aplikácia dP nám uľahčuje i tímovú komunikáciu za pomoci fóra. Túto možnosť však nevyužívame v takej miere, ako iné funkcie, keďže sa ako tím pravidelne a často stretávame. Celkovo bez používania tejto aplikácie by bol manažment nášho tímu o poznanie ťažší a vyžadoval by si oveľa väčšie úsilie. Veď o koľko času sme bohatší, keď nemusíme v mailovej komunikácii hľadať, čo máme spraviť, keď nám prehľad z ganttovho diagramu povie, čo naväzuje na akú úlohu a podobne.

¹ <http://www.dotproject.net/>

6.2 Popis SVN

V našom tíme sme sa rozhodli pre systém na manažment verzií Subversion² (SVN). Táto kapitola obsahuje výhody, ktoré pre náš tím prináša používanie SVN.

SVN, ako už napovedá názov, je systém na správu verzií súborov. Jeho hlavnou črtou je centralizovaná architektúra typu server-klient. Hlavnými výhodami jeho používania je dobrá podpora spolupráce viacerých ľudí na jednom projekte. Tiež umožňuje archivovať vývojové a produkčné verzie projektov. V podstate to je jedno centrálné úložisko súborov, ktoré sa stará o správu verzií týchto dokumentov. Jedna z ďalších výhod je aj jednoduchšie zálohovanie zdrojových kódov, ktoré sú uložené na jednom mieste a nie distribuované medzi členmi tímu.

² <http://subversion.tigris.org/>

Príloha A - Ponuka

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií



Oznamovanie požiarov dobrovoľným hasičom

(Vývoj softvérového systému v tíme - ponuka)

Tím č. 13:

Bc. Matúš Svrček

Bc. Alexander Šimko

Bc. Michal Štekláč

Bc. Miroslav Štolc

Bc. Jaroslav Tešlár

Bc. Ľubomír Varga

Obsah

1 Úvod.....	1
2 Motivácia.....	1
3 Stručný popis funkcií systému.....	2
4 Hrubý návrh.....	3
5 Zloženie tímu.....	4
6 Požadované zdroje.....	5
7 Preferované témy.....	6
8 Emailový kontakt.....	6
9 Rozvrh.....	7

1 Úvod

Oheň sprevádza človeka už odpradáva. Od momentu, keď sa ho naučil využívať vo svoj prospech, predstavuje jeho každodenného spoločníka. Ako sa vraví, je dobrý sluha, no zlý pán. Umožní nám pripravovať pokrm, spracovávať materiál, umožní vozidlám pohyb. Avšak nekontrolovaný oheň môže ničiť a zabíjať.

História ľudstva je popretkávaná mnohými požiarimi, ktoré si vyžiadali obrovské škody. Tak materiálne, ako aj na ľudských životoch. Púť sebazáchovy a zdravý rozum samozrejme ľudí donútil zamýšľať sa nad spôsobmi, ktorými by sa dalo týmto škodám aspoň čiastočne vyhnúť. Najprv si každý ratoval svoj majetok sám. Postupom času sa však človek stal vo svojom boji proti tomuto živlu lepšie pripravený a viac organizovaný. Vznikali skupiny ľudí ochotné brániť a chrániť životy a majetok seba aj ostatných – hasiči.

Hasiči boli ochotní v prípade požiaru využiť dostupné prostriedky na to, aby požiar uhasili a zabránili jeho ďalšiemu šíreniu, prípadne pomáhať ľuďom dostať sa z horiacich objektov. Ich zbraňami bolo spočiatku odhodlanosť, obyčajné vedro s vodou a rýchlosť. V priebehu dejín ponúkal technický pokrok nové možnosti, a tak postupne pribúdali striekačky, najskôr ťahané koňmi a poháňané ľuďmi, neskôr automobilmi. Výbavu hasičov doplnili zdvižné plošiny, vrtuľníky a iná moderná technika.

Technické zariadenia však nie sú v tomto boji to najhlavnejšie. Prvoradé naďalej ostávajú rýchle, presné a dostupné informácie. Pretože práve oni, pri dobrej koordinácii činností a efektívnej spolupráci hasičov, môžu viesť k včasnému zásahu a lepšiemu zvládnutiu vážnej situácie.

2 Motivácia

Azda každý človek vo svojom živote dospeje k bodu, keď si uvedomí, že život nie je len o tom zabávať sa a brať. Rovnakou mierou je potrebné aj dávať. Spôsobov, ktorými môžu ľudia priložiť ruku k dielu, je mnoho. My sme si v našich životoch vybrali softvérové inžinierstvo. To usmerňuje spôsob, ktorým môžeme pomôcť my. Naša sila tkvie v iných schopnostiach, ktoré by v priamom boji zoči voči s plameňom neboli veľmi platné. Chceme pomôcť a usmerniť svoju tvorivú silu smerom, ktorý bude prospešný okoliu a celej spoločnosti. Chceme sa podieľať aspoň malou časťou na spoločnom boji proti nepriateľovi, ktorý je mocnejší než ktokoľvek z nás. Chceme pomôcť pri organizácii a koordinácii zásahových skupín hasičov-dobrovoľníkov. Radi by sme preto v tomto projekte vytvorili systém, ktorý bude tieto činnosti umožňovať.

Na druhej strane je tento projekt pre nás výzvou. Predstavuje reálnu a najmä praktickú funkčnosť, na ktorej nám umožní zdokonaľiť sa vo viacerých technológiách. Mnoho z nich poznáme na úrovni jednoduchého použitia pre testovacie účely, ale teraz bude potrebné riešiť konkrétne požiadavky na výkon, vysokú dostupnosť a podobne. Navyše sa tešíme na samotnú spoluprácu. Sme radi, že si na vlastnej koži vyskúšame, ako funguje rozdelenie a koordinácia činností pri práci na tímovom projekte.

3 Stručný popis funkcií systému

Jednotlivé dobrovoľné požiarne stanice sa budú registrovať na základe pridelenia licencie DPO SR. Následne budú tieto stanice súčasťou systému. Ak sa vyskytne požiar niekde na Slovensku, tak operátor integrovaného záchranného systému oznámi pomocou jednoduchého klienta miesto požiaru. (možnosť automatizovaného hlásenia, ak takéto hlásenie vykonáva v inom systéme) Na základe súradníc tohoto miesta a mapy SR, ktorá bude súčasťou systému, bude vybraná najvhodnejšia stanica alebo stanice. Najvhodnejšia je v zásade najbližšia, resp. tá, ktorá môže najskôr zasiahnuť.

Pri výbere sa môžu uplatňovať ďalšie pravidlá, ktorými sa DPO SR riadi. O ich existencii nemáme momentálne žiadne dostupné znalosti.

V prípade, že by zasahovalo viac zborov naraz, môže operátor na základe informácie o rozsahu požiaru ohodnotiť požiar z hľadiska veľkosti. Na základe tejto veľkosti je následne informovaný príslušný počet zborov hasičov.

Oznámenie hasičskému zboru, čiže informovanie o požiari v jeho blízkosti, môže prebehnúť prostredníctvom mobilnej telekomunikačnej siete a to sms správou alebo telefonátom. Taktiež to môže byť prostredníctvom desktopového klienta na počítači, ktorý je v centrále hasičského zboru, pagerom, alebo iným existujúcim komunikačným prostriedkom.

Spätná väzba od hasičského zboru môže naznačiť, koľko hasičských jednotiek sa dostaví, a teda, či je potrebné oznámiť nutnosť zásahu viacerým jednotkám. Dynamické zmeny požiadavok na hasičskú techniku bude možné operatívne riešiť.

Ako pridanú hodnotu by náš systém mohol obsahovať aj web pre verejnosť so zoznamom všetkých DZZ a ich členov. Druhou alternatívou by bola spolupráca s DPO SR v oblasti verejnej web stránky. Pre riešenie nad rámec projektu by sme boli schopní integrovať do pracovných motorových vozidiel dobrovoľných hasičov sledovacie zariadenie, a tak umožniť operátorovi náhľad na aktuálnu situáciu z vtáčej perspektívy.

4 Hrubý návrh

V tejto sekcii by sme chceli ponúknuť hrubý návrh projektu (technického riešenia), ktorý je ale značne obmedzený malou množinou informácií o konkrétnej problémovej oblasti, resp. o požiadavkách na projekt.

Ponúkame celkový projekt založený na platforme JEE, ktorá patrí v súčasnosti k najpoužívanejším biznis riešeniam na trhu. Aplikácia by pozostávala zo štandardného server-klient riešenia.

Server

Obsahuje celú biznis logiku, ktorá bude umiestnená v aplikačnom serveri ako napríklad JBoss, Apache Geronimo, GlassFish.

Middle

Prenosné (preklápacie) objekty medzi klientom a serverovskými objektami, podľa J2EE vzoru Transfer Object, inak nazývané DTO (data transfer object). Servisná vrstva (podľa J2EE vzoru Service Locator) založená na EJB, a Spring framework-u, v prípade požiadavky na komunikáciu s inými systémami ako Integrovaného Záchranného systému použitie Web Services (cez Apache/Axis).

Táto vrstva obsahuje aj transakčnú časť tvorenú framework-om Hibernate, ktorý zabezpečí aj prepojenie s databázou, kde jednotlivé konekcie sú zaobalené samotným Hibernate.

Databáza, v zásade bude obsahovať údaje o dobrovoľných zboroch a údaje o hlásených požiaroch (preferované sú PostgreSQL, MySQL).

End user

V prípade, že oznamovanie konečným osobám bude prebiehať i cez nami vyvinutý softvér, môžeme dodať desktopové Java aplikácie s plnohodnotným GUI a prípadným signalizačným mechanizmom (zvuková karta PC, sériové i paralelné výstupy, usb). Grafické rozhranie by bolo navrhované ako tenký klient založený na JSP alebo JSF.

5. Zloženie tímu

Sme presvedčení, že pre zadanú tému sme vhodnými riešiteľmi. Náš tím sa skladá z ľudí ktorí pracujú/pracovali v komerčnej oblasti, a majú skúsenosti s prácou v tíme. Naše zloženie pokrýva jednotlivé problémové oblasti, vrstvy architektúry, či technológie.

Matúš Svrček

Je absolventom bakalárskeho štúdia v odbore Informatika na FIIT. V súčasnosti pracuje ako administrátor. Ma viacročnú prax v oblasti správy systémov. Ovláda jazyky Java, C, C++, C#, Delphi, Bash. Sústreďuje sa najmä na jazyk Java, v ktorom vytvoril aj svoju bakalársku prácu zaoberajúcu sa evolučnými algoritmami. Má skúsenosti s hlásením chýb pomocou SMS správ. Ovláda UML.

Alexander Šimko

V rokoch 2006 a 2007 pracoval vo firme Monogram Technologies s.r.o. (www.monogram.sk) na pozícií web developer. Pracovné zameranie bola hlavne prezentačná vrstva. Počas svojej práce v tejto firme sa podieľal na vývoji dvoch veľkých projektoch:

- web aplikácie vyhľadávania leteniek a ubytovaní nasadených pre:
 - České Aerolínie (flight-hotel.czechairlines.com)
 - Železničnú spoločnosť slovensko (www.slovakrail.sk)
- web aplikácia zberu a vyhodnocovania dotazníkov pre spoločnosť Armstrong Competence Consulting

Za rok 2006 bol firmou ocenený za najlepšie zdokonalenie sa.

Bakalárske štúdium ukončil na FIIT STU v Bratislave s udelením diplomu Magna Cum Laude za vynikajúce študijné výsledky.

V súčasnosti pracuje rovnako v oblasti webu, ovláda:

C, C++, C#, ASP.NET (v Monogram Technonogies) , PHP, Java, JEE, Java Server Faces (súčasnosť), SQL, XML, XHTML, JavaScript, CSS, práca s GNU/Linux, Windows

Michal Štekláč

Bakalárske štúdium ukončil na Fakulte Informatiky a Informačných technológií STU v odbore Informatika. Skúsenosti z praxe zatiaľ nemá.

Počítačové znalosti: C, C++, Java, HTML, UML, Linux, Windows.

Miroslav Štolc

Je absolventom bakalárskeho štúdia na Fakulte Informatiky a Informačných technológií v odbore Informatika. Má prax vyše poldruha roka vo veľkej softvérovej firme na pozícii programátor. Sústreďuje sa na vývoj na platforme Java. Má skúsenosti s rôznorodými technológiami, ktoré sa používajú na vývoj J2EE aplikácií. A to konkrétne Apache (a jeho rôzne podprojekty ako napr. Tomcat, Axis), JBoss, JDBC, objektovo relačný framework Hibernate, WebServices, EJB, Spring. Je pokročilým používateľom vývojového prostredia Eclipse, ovláda jeho architektúru pluginov a časť vnútornej reprezentácie jednotlivých objektov Eclipse. Medzi jeho posledné oblasti skúmania patria práve frameworky pod záštitou Eclipse, a to Eclipse Modelling Framework, Graphical Modelling Framework. Ovláda notáciu UML na pokročilej úrovni, má skúsenosti s tvorbou jednotlivých diagramov.

Jaroslav Tešlár

Absolvoval bakalársky stupeň vysokoškolského štúdia v študijnom odbore Informatika na Fakulte infromatiky a informačných technológií STU. Od júla 2007 pracuje v spoločnosti Cosmotron. Venuje sa vývoju databázových aplikácií postrelačného systému Caché. Z programovania ďalej ovláda: C, Java, J2EE (JSP), JavaScript + Google Maps API.

Ľubomír Varga

Rozsiahle skúsenosti v rôznych oboroch. Pôsobil jeden rok v automatizárskej firme ako programátor desktopových aplikácií v .NET, kde prehĺbil svoje elektrotechnické a strojárske znalosti nadobudnuté na strednej škole. Bakalársku prácu vypracoval na platforme Java a ďalej pokračuje vo vývoji SW pod krídlami Javy. Skúsenosti s GPS a GPRS zariadením určeným na online sledovanie pohybu motorových vozidiel. V oblasti serverov má praktické skúsenosti s WMS (web map server) a PostGIS na Postgresql, ako i programovanie vlastných služieb komunikujúcich cez Internet na úrovni soкетов.

6 Požadované zdroje

Naše preferované prostredie je Java, konkrétne platforma JEE. Čo sa týka jednotlivých podsystémov preferujeme OpenSource projekty ako Apache, GlassFish, JBoss, Hibernate, PostgreSQL, MySQL, JSF, Ajax4JSF, RichFaces. Výhodou týchto riešení je ich voľná dostupnosť.

Jedinou nevyhnutnou požiadavkou na softvér (hardvér) je SMS brána alebo dohoda s mobilným operátorom, ktorý by poskytol vývojové prostriedky na odosielanie SMS správ, prípadne volaní. (možnosť i „dummy“ serveru)

Požiadavky na miestnosti nemáme. Máme k dispozícii server na vývoj a testovanie. Na tomto serveri je nasadených niekoľko aplikácií, ktoré využívame na organizáciu a riadenie nášho tímu:

- DotProject (web nástroj na riadenie projektov a tímov)
- DokuWiki (projektové dokumentácie, know-how)
- SVN (nástroj na správu verzií)
- TeamSpeak (hlasová komunikácia)

Ďalej sa tam nachádzajú aplikácie na vývoj:

- Tomcat (Java web servlet container)
- Apache (Web server)
- viacero druhov databáz (PostgreSQL, MySQL)

Po vytvorení aplikácie bude potrebné dodať server, na ktorom bude riešenie nasadené.

7 Preferované témy

1. Oznamovanie požiarov dobrovoľným hasičom
2. Simulácia a vizualizácia osvetlenia budov pre architektov so zameraním na samostatne stojace budovy v horách
3. Distribuovaný systém na riešenie symetrickej hry
4. Tvorba testov s využitím LaTeXu

8 Emailový kontakt

shanki@ynet.sk

9 Rozvrh

	07:00	08:00	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	
Pondelok															
Utorok															
Streda															
Štvrtok															
Piatok															

Distribuované riešenie symetrickej hry

A - 8

škola

Príloha B – Zápisy zo stretnutí

Zápis zo stretnutia tímu č. 13 [Švábi]

Stretnutie číslo: 1
 Dátum a čas: 9.10.2007, 7:30 - 9:15
 Miesto: Softvérové štúdio
 Zúčastnení členovia tímu: Bc. Michal Štekláč, Bc. Matúš Svrček, Bc. Lubomír Varga,
 Bc. Alexander Šimko, Bc. Jaroslav Tešlár, Bc. Miroslav Štolc
 Vedúci projektu: Ing. Peter Lacko
 Hostia: -

Téma stretnutia

Prvé stretnutie tímu spolu s vedúcim projektu. Najprv bola stručne zhodnotená ponuka tímu. Prebehla konkretizácia zadania a boli identifikované hlavné problémy - prvotné úlohy pre členov tímu.

Priebeh stretnutia

Zhodnotenie prebiehajúcich úloh

Jednalo sa o prvé stretnutie, neexistovali prebiehajúce úlohy.

Samotné stretnutie

Bol prednesený stručný popis prostredia BOINC (<http://boinc.berkeley.edu>) na ktorom bude projekt nasadený. Ten pozostáva z dvoch častí klient server.

Zásadné stanoviská:

- projekt má byť modulárny t.j. použiteľný pre rôzne distribuované hry
- dôležitosť návrhu pracovnej jednotky (work unit) pre klientskú stranu a stratégie rozdeľovania úloh pre klientov

Aktuálny plán úloh

Člen tímu	Úloha	Termín začiatku	Termín ukončenia
Matúš	Nasadenie softvéru BOINC, štúdium alfa-beta osekávania	9.10.2007	13.10.2007
Lubomír	Analýza hry GO, prvotný návrh klienta	9.10.2007	13.10.2007
Michal	Analýza hry Reversi	9.10.2007	13.10.2007
Alexander	Algoritmus riešenia, preskúmanie existujúcich	9.10.2007	13.10.2007
Jaroslav	Návrh webovskej stránky tímu	9.10.2007	13.10.2007
Miroslav	Vytvorenie šablón pre tímové dokumenty	9.10.2007	12.10.2007
Všetci	Návrh pre work unit	9.10.2007	13.10.2007

Príloha B

Tím číslo 13: Švábi

Zapísoval:

Bc. Miroslav Štolc

Zápis zo stretnutia tímu č. 13 [Švábi]

Stretnutie číslo: 2
 Dátum a čas: 16.10.2007 7:00 – 10:00
 Miesto: Softvérové štúdio
 Zúčastnení členovia tímu: Bc. Michal Štekláč, Bc. Matúš Svrček, Bc. Lubomír Varga,
 Bc. Alexander Šimko, Bc. Jaroslav Tešlár, Bc. Miroslav Štolc
 Vedúci projektu: Ing. Peter Lacko
 Hostia: -

Téma stretnutia

Na druhom stretnutí tímu sa prezentovali v skratke dosiahnuté výsledky analýz a následne sa analyzovalo ďalej. Stručne sa prebral algoritmus alfa-beta osekávania a jeho možné implementácie v distribuovanom prostredí a načrtli sa možné riešenia ukladania stavového grafu problému na disk.

Priebeh stretnutia

Zhodnotenie prebiehajúcich úloh

Koordinátor stretnutia na začiatku stretnutia postupne predstavil riešené problémy a ich riešiteľov, ktorí uviedli výsledky svojich analýz a praktických pokusov.

Člen tímu	Úloha	Termín začiatku	Termín ukončenia	Splnená
Michal	Predstavenie výsledkov Reversi	9.10.2007	13.10.2007	Áno
Matúš	Predstavenie okruhu algoritmov vyhľadávania a predstavenie nainštalovaného boinc serveru	9.10.2007	13.10.2007	Áno
Lubomír	Predstavenie výsledkov GO a predstavenie boinc platformy	9.10.2007	13.10.2007	Áno
Jaroslav	Vytvorenie webu	9.10.2007	13.10.2007	Áno
Alexander	Skúmanie algoritmov	9.10.2007	13.10.2007	Áno
Miroslav	Šablóny dokumentov	9.10.2007	13.10.2007	Áno
Všetci	Návrh pre workunit (potrebná ďalšia analýza oblasti)	9.10.2007	13.10.2007	Áno

Samotné stretnutie

Popis prednesených výsledkov nieje potrebné uvádzať, ich obsah je v dotyčných dokumentoch

v tíme.

Ujasnilo sa, že výsledkom nášho skúmania (projektu) nemusí byť strom riešení, ale ako postačujúci (ultra-weak) výsledok je i odpoveď, či existuje výherná stratégia pre bieleho (začínajúceho) hráča. Ak neexistuje, bolo by vhodné zistiť pomer výhier bieleho a čierneho.

Veľkou časťou diskusie bolo rozoberanie fungovania algoritmu a výpočtu v distribuovanom prostredí. Dospelo sa k záveru, že bude nutné preštudovať viac materiálov k danej téme, čo sa odrazilo v pláne úloh. Navrhnuté riešenia boli funkčné, avšak bola diskutovaná ich efektívnosť.

Pri diskusii o spôsobe výpočtu a rozdeľovania práce sa dospelo k otázke, koľko pamäťového priestoru budeme potrebovať pre udržiavanie si stromu. Po rôznych odhadoch sa dospelo k potrebe 128 bitov na reprezentáciu šachovnice a potrebu niekoľkých ukazovateľov. Jeden minimálne na rodiča (i keď môže vzniknúť situácia, že do jedného stavu sa dá dostať dvoma spôsobmi, toto budeme zanedbávať a za pomoci transpozičnej tabuľky priamo ohodnotíme neskoršie vygenerovaný stav a necháme ho v pamäti dvojmo) a niekoľko potomkov a/alebo susedov. Presnejší spôsob uloženia ukázu potrebné analýzy.

Aktuálny plán úloh

Člen tímu	Úloha	Termín začiatku	Termín ukončenia
Matúš	Preštudovať možnosti nejakých filesystémov pre ukládanie stromu stavov	16.10.2007	22.10.2007
Matúš	vytvoriť konto pre vedúceho projektu	16.10.2007	22.10.2007
Matúš, Miroslav, Michal, Jaroslav, Alexander	Preštudovať v ACM knižnici nejaké algoritmy pre alfa-beta prehľadávanie a iné vylepšenia	16.10.2007	29.10.2007
Jaroslav	Pridávať pravidelne zápisy zo stretnutia na web	16.10.2007	priebežne
Ľubomír	Rozbehať skúšobný sample boinc projekt	16.10.2007	22.10.2007
Alexander	Pridať tému ACM algoritmy, do fóra na dotProject-e	16.10.2007	22.10.2007

Zapísal:

Bc. Ľubomír Varga

Zápis zo stretnutia tímu č. 13 [Švábi]

Stretnutie číslo: 3
 Dátum a čas: 23.10.2007 7:00 – 10:00
 Miesto: Softvérové štúdio
 Zúčastnení členovia tímu: Bc. Michal Štekláč, Bc. Matúš Svrček, Bc. Lubomír Varga,
 Bc. Alexander Šimko, Bc. Jaroslav Tešlár, Bc. Miroslav Štolc
 Vedúci projektu: Ing. Peter Lacko
 Hostia: -

Téma stretnutia

Načrtli sa ďalšie výsledky analýzy – nové heuristiky, výhody a nevýhody použitia databázy alebo súborového systému na uloženie MINIMAX stromu na strane servera.

Priebeh stretnutia

Zhodnotenie prebiehajúcich úloh

Koordinátor stretnutia na začiatku stretnutia postupne predstavil riešené problémy a ich riešiteľov, ktorí uviedli výsledky svojich analýz a praktických pokusov.

Člen tímu	Úloha	Termín začiatku	Termín ukončenia	Splnená
Matúš, Miroslav, Michal, Jaroslav, Alexander	Preštudovať v ACM knižnici nejaké algoritmy pre alfa-beta prehľadávanie a iné vylepšenia	16.10.2007	29.10.2007	Čiastočne
Matúš	Preštudovanie možností použitia filesystémov alebo databázových riešení na uloženie minimax stromu na serveri	16.10.2007	22.10.2007	Áno
Alexandert	vytvoriť konto pre vedúceho projektu na dotProjecte a pridať tému ACM algoritmy do fóra	16.10.2007	22.10.2007	Áno
Jaroslav	úprava webu, doplnenie plánu a zápisov zo stretnutí vo formáte HTML	16.10.2007	22.10.2007	Áno
Člen tímu*	Úloha	Termín začiatku	Termín ukončenia	Splnená
Lubomír	Rozbehať skúšobný sample boinc projekt	16.10.2007	22.10.2007	Nie

Samotné stretnutie

Po Matúšovom zhodnotení uloženia MINIMAX stormu na strane servera sa dospelo k tomu, že bude použitá databáza PostgreSQL. Jej hlavnou výhodou je, že dokáže ukladať aj 16-bajtové údaje, ktoré dokážu reprezentovať stav jednej konkrétnej šachovnice. Navyše poskytuje obrovské priestory na ukladanie dát. Naproti tomu, existujúce filesystemy sú obmedzené najmä veľkosťou najmenšieho súboru 512 bajtov, čo by znamenalo na reprezentáciu jedného uzla použiť 32-krát viac miesta, ako je potrebné. K dispozícii bude diskové pole o veľkosti asi 1 TB, preto bude treba zvážiť, koľko úrovní stromu bude na serveri uložených (približne 10).

Ďalej sa predniesli nové možné heuristiky použiteľné pre generovanie stromu. Táto fáza ešte nie je na konci, je potrebné preštudovať ich ešte viac a vhodné vybrať. Vedúci tímu dal pokyny, aby sa pri návrhu uvažovalo o modularite, teda aby generovanie ťahov pre konkrétnu hru bolo samostatným a celý systém bol univerzálnejší na použitie pre iné symetrické hry.

Aktuálny plán úloh

Člen tímu	Úloha	Termín začiatku	Termín ukončenia
Michal	vytvoriť template pre dokumentáciu	23.10.2007	29.10.2007
Miroslav	programové konvencie pre projekt, skúsiť nejaké štatistiky pre tvorbu stromu reversioni na jednom počítači (faktor vetvenia, doba trvania...)	23.10.2007	29.10.2007
Matúš, Miroslav, Michal, Jaroslav, Alexander	Preštudovať v ACM knižnici nejaké algoritmy pre alfa-beta prehľadávanie a iné vylepšenia (pokračovanie)	16.10.2007	29.10.2007
Jaroslav	Pridávať pravidelne zápisy zo stretnutia na web	16.10.2007	priebežne
Ľubomír	Rozbehať skúšobný sample boinc projekt	23.10.2007	29.10.2007
	drobný návrh, aké moduly a podobne	23.10.2007	29.10.2007

Zapísal:

Bc. Jaroslav Tešlár

Zápis zo stretnutia tímu č. 13 [Švábi]

Stretnutie číslo: 4
 Dátum a čas: 30. 10. 2007
 Miesto: Softvérové štúdio
 Zúčastnení členovia tímu: Bc. Michal Štekláč, Bc. Matúš Svrček, Bc. Lubomír Varga,
 Bc. Alexander Šimko, Bc. Jaroslav Tešlár, Bc. Miroslav Štolc
 Vedúci projektu: Ing. Peter Lacko
 Hostia: -

Téma stretnutia

Hlavnou témou štvrtého stretnutia bolo vytvorenie prvotného návrhu. Na stretnutí sa pokračovalo v analyzovaní problémovej oblasti.

Priebeh stretnutia

Zhodnotenie prebiehajúcich úloh

Na začiatku boli vyhodnotené zadané úlohy z tretieho stretnutia. Jednotliví členovia prezentovali ako pokročili v zadaných úlohách.

Člen tímu	Úloha	Termín začiatku	Termín ukončenia	Splnená
Miroslav	programové konvencie pre projekt, skúsiť nejaké štatistiky pre tvorbu stromu reversi na jednom počítači (faktor vetvenia, doba trvania...)	23. 10. 2007	29. 10. 2007	Áno
Jaroslav	pridávať pravidelne zápisy zo stretnutia na web	16. 10. 2007	priebežne	Áno
Lubomír	rozbehať skúšobný sample boinc projekt	23. 10. 2007	29. 10. 2007	Áno
Všetci	drobný návrh, aké moduly a podobne	23. 10. 2007	29. 10. 2007	Áno
Michal	vytvoriť template pre dokumentáciu a celková úprava dokumentácie	23. 10. 2007	29. 10. 2007	čiastočne

Samotné stretnutie

Na stretnutí Lubomír v krátkosti zreferoval o rozbehaní jednoducho príkladu v boincu. Ten bol rozbehnutý na jednom počítači z čoho vyplynula ďalšia úloha vyskúšať tento príklad na viacerých počítačoch.

Miroslav nás oboznámil aký je faktor vetvenia pre hre reversi. Faktor vychádza na začiatku a

konci hry okolo 6. V strede hry sa faktor vetvenia pohybuje okolo 9. Tieto faktory vetvenia boli získané z implementácie, ktorú Miroslav implementoval.

Ďalším bodom stretnutia bolo vytvorenie počiatočného návrhu. Identifikovali sme jednotlivé funkcie, ktoré boli následne rozdelené medzi serverovú a klientskú časť a určili sme závislosť jednotlivých funkcií od hry. Návrh bol zdokumentovaný, bude sa z neho vychádzať v ďalšej práci.

Posledným bodom bolo rozdelenie úloh pri písaní dokumentácie.

Aktuálny plán úloh

Člen tímu	Úloha	Termín začiatku	Termín ukončenia
Ľubomír	rozbehať skúšobný sample boinc projekt na viacerých počítačoch	30. 10. 2007	5. 11. 2007
Miroslav	vytvorenie konvencií ako písať zdrojové kódy	30. 10. 2007	9. 11. 2007
Jaroslav	dokumentácia – o hrách	30. 10. 2007	5. 11. 2007
Michal	dokumentácia - opísanie existujúcich riešení, špecifikácia, slovník pojmov - vytvoriť	30. 10. 2007	5. 11. 2007
Ľubomír	dokumentácia – o boinc-u	30. 10. 2007	5. 11. 2007
Matúš	dokumentácia – možnosti ukladania stromu	30. 10. 2007	5. 11. 2007
Alexander	dokumentácia – opísanie potrebných algoritmov	30. 10. 2007	5. 11. 2007
Miroslav	dokumentácia – opísanie hrubého návrhu a prototypu	30. 10. 2007	5. 11. 2007

Zapísal:

Bc. Michal Štekláč

Zápis zo stretnutia tímu č. 13 [Švábi]

Stretnutie číslo: 5
 Dátum a čas: 06. 11. 2007
 Miesto: Softvérové štúdio
 Zúčastnení členovia tímu: Bc. Michal Štekláč, Bc. Matúš Svrček, Bc. Ľubomír Varga,
 Bc. Alexander Šimko, Bc. Jaroslav Tešlár, Bc. Miroslav Štolc
 Vedúci projektu: Ing. Peter Lacko
 Hostia: -

Téma stretnutia: príprava projektovej dokumentácie

Priebeh stretnutia

Zhodnotenie prebiehajúcich úloh

Na začiatku boli vyhodnotené zadané úlohy zo štvrtého stretnutia. Jednotliví členovia prezentovali, ako pokročili v zadaných úlohách.

Michal pracoval na dokumentácii. Miroslav sa ďalej venoval stromom hry. Ostatní členovia tímu sa ďalej venujú prácam na dokumentácii.

Člen tímu*	Úloha	Termín začiatku	Termín ukončenia	Stav
Ľubomír	rozbehať skúšobný sample boinc projekt na viacerých počítačoch	30. 10. 2007	5. 11. 2007	ukončené
Miroslav	vytvorenie konvencií ako písať zdrojové kódy	30. 10. 2007	5. 11. 2007	čiastočne
Jaroslav	dokumentácia – o hrách	30. 10. 2007	5. 11. 2007	čiastočne
Michal	dokumentácia - opísanie existujúcich riešení, špecifikácia, slovník pojmov - vytvoriť	30. 10. 2007	5. 11. 2007	čiastočne
Ľubomír	dokumentácia – o boinc-u	30. 10. 2007	5. 11. 2007	čiastočne
Matúš	dokumentácia – možnosti ukladania stromu	30. 10. 2007	5. 11. 2007	čiastočne
Alexander	dokumentácia – opísanie potrebných algoritmov	30. 10. 2007	5. 11. 2007	čiastočne
Miroslav	dokumentácia – opísanie hrubého návrhu a prototypu	30. 10. 2007	5. 11. 2007	čiastočne

Samotné stretnutie a plán úloh

Venovali sme sa hlavne dokumentácii, keďže termín jej odovzdania je už budúce stretnutie. Presnejšie sme definovali, ktoré časti dokumentácia bude obsahovať, a rozdelili sme si čiastkové úlohy pri jej príprave. Alexander nás informoval, že prideliť jednotlivým členom ich úlohy.

Na stretnutí sme sa zhodli na:

- V dokumentácii bude potrebné uviesť časť zaoberajúcu sa riadením
- Bude treba vypracovať preberací protokol
- V dokumentácii minulých tímov zaberalo riadenie zhruba polovicu z celej dokumentácie (teda cca 30 strán)
- Hlavnou úlohou na budúce stretnutie je pripraviť dokumentáciu, pridelenie ľudí na čiastkové úlohy je takéto:

Člen tímu*	Úloha	Termín začiatku	Termín ukončenia
Všetci	Podrobný plán tímového projektu	5. 11. 2007	10. 11. 2007
Michal	Rozdeľovanie úloh	5. 11. 2007	10. 11. 2007
Jaroslav	Záznamy zo stretnutí	5. 11. 2007	10. 11. 2007
Miroslav	Štandardy použité pri zdrojovom kóde	5. 11. 2007	10. 11. 2007
Matúš	Manažment verzií konfigurácií a zmien	5. 11. 2007	10. 11. 2007
Michal	Preberacie protokoly	5. 11. 2007	10. 11. 2007
Alexander	Komunikácia v tíme	5. 11. 2007	10. 11. 2007
Lubomír	Popis dotProjectu	5. 11. 2007	10. 11. 2007
Matúš + Jaroslav	Zverejnenie zápisov na web stránke	5. 11. 2007	10. 11. 2007
Michal	Šablóna na slovník pojmov	5. 11. 2007	10. 11. 2007
Michal	Šablóna pre dokumentáciu a konečná úprava dokumentácie	5. 11. 2007	10. 11. 2007
Matúš	Záverečná kontrola dokumentácie – pravopis a formátovanie	5. 11. 2007	10. 11. 2007
Michal	Autorstvo	5. 11. 2007	10. 11. 2007
Michal	Úlohy členov tímu	5. 11. 2007	10. 11. 2007

- Termín na splnenie týchto úloh je sobota 10.11.2007, dokumentáciu odovzdáme vedúcemu projektu na posúdenie

Zapísal: Bc. Matúš Svrček

*členovia tímu sú udávaní krstným menom, ktoré je v tíme jedinečné

Príloha C - Programátorské konvencie

Programátorské konvencie

Konvencie sú určené pre programátorov s cieľom čo najviac zefektívniť vývoj produktu. Dodržiavanie týchto konvencií patrí medzi základné programátorské zručnosti. Konvencie tvorby kódu sú dôležité z viacerých príčin: sprehľadňuje zdrojový kód, uľahčuje údržbu a zdieľanie rôznych súborov medzi programátormi. Jazyk pre identifikátory nieje súčasťou týchto konvencií, je to manažérske alebo zákaznícke rozhodnutie, ale jednoznačne sa odporúča písať zdrojový kód v jednom jazyku. Tieto konvencie sú pre vývoj v jazyku C. Tento dokument obsahuje popis použitia kľúčových slov jazyka C, tie budú pre prehľadnosť písané tučne (napr. **return**).

Súbory

V programe napísanom v jazyku C existujú dve skupiny súborov, klasický zdrojový súbor s príponou .c a hlavičkový (deklaračný) súbor s príponou .h. Mená súborov sú písané malou abecedou, ak názov súboru obsahuje dve slová, tak sú oddelené podčiarkovníkom "_". Súbory obsahujúce viac ako 1000 riadkov sa považujú za predimenzované, je odporúčané ich rozdeliť podľa funkčnosti. Jeden riadok súboru musí byť viditeľný na obrazovke, treba sa vyhýbať dlhým príkazom, ak presahujú 80 znakov sa logicky rozdeliť na ďalšie riadky do 80 znakov a od druhého riadku odsadené tabulátorom.

Organizácia súborov

Zdrojové súbory sú organizované nasledovne:

- úvodný komentár
- deklarácie rozhraní
- definície konštánt
- definície vlastných typov
- deklarácie funkcií
- definície funkcií

Organizácia hlavičkového súboru je identická až na absenciu poslednej časti - definície funkcií.

Úvodný komentár

Je doporučený najmä pri projektoch, ktoré obsahujú viac ako dvadsať súborov, je typu blokový komentár. Komentár sa skladá:

- meno súboru
- stručný popis funkcionality
- dátum poslednej úpravy

- meno autora

Deklarácie rozhraní

Pod rozhraním rozumieme **#include** príkazy, čiže referenciu na hlavičkový súbor. Najprv štandardné potom vlastné knižnice oddelené prázdny riadkom, zoradené podľa abecedy.

Definícia konštánt

V jazyku C cez vyhradené slovo **#define**, konštanta je písaná výhradne veľkou abecedou slová oddelené podčiarkom.

Definícia vlastných typov

Pod typom v jazyku rozumieme štruktúru (**struct** alebo **union**) a enumeráciu (**enum**). Definícia začína slovom **typedef**, nasleduje typ a meno. Meno typu je písané klasickou abecedou, slová začínajú s veľkým písmenom, napr.:

```
typedef struct PrvokZoznamu {  
    //jednotlivé položky  
} PrvokZoznamu;
```

Deklarácie/definície funkcií

Funkcia je deklarovaná podľa pravidiel jazyka C, meno funkcie pozostáva z písmen abecedy, slová začínajú veľkým písmenom, prvé písmeno z mena je malé. Funkcie sú zoradené logicky podľa funkcionality, nie podľa abecedy. Príklad:

```
int zoradUlohy();
```

resp.

```
int zoradUlohy() {  
    //telo funkcie  
}
```

Definície funkcií v tom poradí ako boli funkcie deklarované, ak súbor obsahuje špeciálnu funkciu `main()`, tá je umiestnená ako prvá hneď po deklaráciách funkcií.

Komentáre

Komentáre môžeme rozdeliť do dvoch skupín:

- blokový komentár
- jednoriadkový komentár

Blokový komentár

Blokové komentáre sa používajú na popis súborov, metód, údajových štruktúr a algoritmov. Blokové komentáre by sa mali používať na začiatku súboru a pred metódou. Samozrejme môžu sa použiť kdekoľvek napr. i priamo v metóde. Blokový komentár vo vnútri metódy by mal byť obsadený na tej úrovni ako kód, ktorý popisuje a mal by byť štruktúrovaný minimálne do viet, odporúča sa použiť odrážky. Príklad:

```
/*  
 * Toto je blokovy komentar.  
 */
```

Jednoriadkový komentár

Jednoriadkový komentár je reprezentovaný oddeľovačom "//", ktorý je možné použiť pre okomentovanie celého riadku alebo časti riadku. Nemal by sa používať pre vytvorenie viac riadkových komentárov. Príklad:

```
// toto je jednoriadkovy komentar
```

Definície/Deklarácie

Premenné, inicializácia

Premenné sú doporučene deklarované v jednom riadku najlepšie s popisným komentárom:

```
int pocetPoloziek; // pocet poloziek na fakture
```

Nikdy nedeklarovať v jednom riadku dva rôzne typy:

```
int pocet, pole[];
```

Inicializujte lokálne premenné, tam kde sú deklarované. Jedinou možnou príčinou prečo tak neurobiť na mieste deklarácie je potreba výpočtu počiatočnej hodnoty.

Smerník je umiestnený pri mene premennej nie pri type:

```
int *smernikNaInt;
```

Umiestenie bloku

Blok je presne definovaná časť zdrojového kódu, ktorá je ohraničená špeciálnymi zátvorkami "{ telo bloku }". Otváracia zátvorka "{" patrí na koniec riadku deklarácie. Uzatváracia zátvorka "}" patrí na začiatok nového riadku odsadená na rovnakej úrovni ako korešpondujúci otvárací príkaz. Príklad:

```
if (podmienka) {  
    // príkazy  
}
```

Funkcie

Žiadna medzera medzi menom metódy a zátvorkou pre vstupné parametre. Telo funkcie je blok, jeho umiestenie sa riadi podľa toho.

Príkazy

Každý riadok obsahuje maximálne jeden príkaz. Vyhradené slovo je vždy od zátvorky oddelené práve jednou medzerou.

Príkazy if, if-else, if-else-if

Triedy príkazov **if** majú nasledujúci tvar:

```
if (podmienka) {
    // príkazy
}

if (podmienka) {
    // príkazy
}
else {
    // príkazy
}

if (podmienka) {
    // príkazy
}
else if (podmienka) {
    // príkazy
}
```

Príkazy cyklu

Príkazy cyklu majú nasledovaný tvar:

```
for ( inicializačná časť; podmienka; časť zmeny riadiacich premenných cyklu) {
    // príkazy
}

while (podmienka) {
    // príkazy
}

do {
    // príkazy
} while (podmienka);
```

Príkaz výberu - switch

Príkaz **switch** má tvar:

```
switch (výraz) {
    case ABC:
        príkazy;
        // vykonávanie prejde na ďalšiu množinu príkazov pre prípad DEF
    case DEF:
        // príkazy
        break;
```

```
default:  
    príkazy;  
    break;  
}
```

Programovacie postupy

Ďalšie postupy podľa ktorých sa má riadiť programátor pri písaní príkazov:

- operátory =, ==, !=, +, -, *, /, %, &&, ||, &, |, << a >> sú pred a aj za oddelené medzerou
- príkaz priradenia = je použitý v jednom riadku práve raz
- pretypovanie by mala nasledovať medzera: (byte) premenna;
- prístup k štruktúrovaným premenným je bez medzere: osoba->meno resp. osoba.meno

Príloha D – Percentuálne rozdelenie bodov

Percentuálne rozdelenie bodov

Bc. Matúš Svrček	16,6 %
Bc. Alexander Šimko	16,6 %
Bc. Michal Štekláč	16,6 %
Bc. Miroslav Štolc	16,6 %
Bc. Jaroslav Tešlár	16,6 %
Bc. Ľubomír Varga	16,6 %

Príloha E - Preberacie protokoly

Preberací protokol k projektovej dokumentácii

Autor:

Názov tímu: **Švábi**

Členovia tímu: Bc. Matúš Svrček,
Bc. Alexander Šimko,
Bc. Michal Štekláč,
Bc. Miroslav Štolc,
Bc. Jaroslav Tešlár,
Bc. Lubomír Varga

Posudzovateľ:

Ing. Peter Lacko

Dolupodpísaný posudzovateľ svojím podpisom potvrdzuje, že prebral od autora dokumentáciu k tímovému projektu, pozostávajúcu z dokumentácie k projektu v rozsahu strán a z riadenia projektu v rozsahu strán s prílohami.

Dátum

.....

Posudzovateľ

Autor:

.....

Ing. Peter Lacko

.....

Bc. Alexander Šimko
za tím Švábi

Preberací protokol k projektovej dokumentácii

Autor:

Názov tímu: **Švábi**

Členovia tímu: Bc. Matúš Svrček,
Bc. Alexander Šimko,
Bc. Michal Štekláč,
Bc. Miroslav Štolc,
Bc. Jaroslav Tešlár,
Bc. Lubomír Varga

Posudzovateľ:

Názov tímu:.....

Dolupodpísaný posudzovateľ svojím podpisom potvrdzuje, že prebral od autora dokumentáciu k tímovému projektu, pozostávajúcu z dokumentácie k projektu v rozsahu strán a z riadenia projektu v rozsahu strán s prílohami.

Dátum

.....

Posudzovateľ

Autor:

.....
Bc.
za tím.....

.....
Bc. Alexander Šimko
za tím Švábi