



## *Robocup S*

Práca s vývojovým prostredím, riadenie verzií a štandardy kodovania

verzia 0.2

## História zmien

---

<b>Dátum zmeny</b>	<b>Verzia dokumentu</b>	<b>Popis zmeny</b>	<b>Autor</b>
16.12.2006	0.1	Založenie dokumentu	Martin Kováčik
18.12.2006	0.2	Pridanie štandardov kódovania	Peter Kohaut

## **Anotácia**

---

Cieľom tohto dokumentu je definovať metodiky, potrebné pre vývoj hráča tými GangOfSix. Dokument popisuje:

- Všeobecné informácie o využívaných prostriedkoch a referencie uvádzané v dokumente
- Inicializáciu pracovného prostredia
- Štruktúra pracovného prostredia a pridávanie nových súborov
- Správa verzií
- Štandardy kódovania

Dokument je členený do kapitol, kde každá kapitola reprezentuje uvedený tematický celok.

## Obsah dokumentu

---

1 ÚVOD.....	I
1.1 Terminológia.....	I
2 ŠPECIFIKÁCIA PROSTRIEDKOV POUŽÍVANÝCH NA VÝVOJ.....	II
2.1 Zoznam systémov, ktoré tvoria vývojové prostredie.....	II
2.2 Zoznam nastavení prostredia.....	II
3 INICIALIZÁCIA PRACOVNÉHO PROSTREDIA.....	III
3.1 Inštalácia vývojového prostredia.....	III
3.1.1 Inštalácia systému Subversion a Mingw:.....	III
3.1.2 Inštalácia prostredia Eclipse a potrebných rozšírení.....	III
3.1.3 Nastavenie premennej PATH.....	III
3.2 Nastavenie prístupu k SVN a stiahnutie zdrojových kódov.....	IV
4 ŠTRUKTÚRA PRACOVNÉHO PROSTREDIA.....	VI
4.1 Adresárová štruktúra projektu.....	VI
5 SPRÁVA VERZIÍ.....	VII
5.1 Vytvorenie novej vetvy.....	VII
5.2 Štandardy zápisu komentárov v repozitári.....	VII
5.3 Ukončovanie vývoja na vetve.....	VIII
5.4 Spájanie vetiev.....	VIII
6 ŠTANDARDY PÍSANIA KÓDU.....	X
6.1 Vizualna stránka kódu.....	X
6.2 Pravidlá pre použitý jazyk.....	XI
PRÍLOHY.....	XII
A. Bibliografia.....	XII
B. Zoznam tabuliek.....	XII
C. Zoznam obrázkov.....	XII

# 1 Úvod

---

## 1.1 Terminológia

Pojem/Skratka	Vysvetlenie
Repository	Repositár. Ide o úložisko využívané systémom Subversion. Oba pojmy sú ekvivalentné.

## 2 Špecifikácia prostriedkov používaných na vývoj

### 2.1 Zoznam systémov, ktoré tvoria vývojové prostredie

System	Opis použitia	Povinné	Referencia
Eclipse 3.2	Hlavné vývojové prostredie. Umožňuje integráciu s kompilátorom Mingw a poskytuje rozhranie pre prácu s verziami.	nie	<a href="http://www.eclipse.org">www.eclipse.org</a>
Eclipse CDT	Plugin do Eclipse. Obsahuje podporu pre prácu s C++ projektami	nie	<a href="http://www.eclipse.org">www.eclipse.org</a>
Subclipse	Plugin do Eclipse. Obsahuje podporu pre prácu s Subversion.	nie	<a href="http://subclipse.tigris.org">http://subclipse.tigris.org</a>
Mingw	Set nástrojov obsahujúci kompilátor, nástroje podporujúce automatizovanú kompiláciu a knižnice	áno	<a href="http://www.mingw.org">www.mingw.org</a> Distribučný balík zahŕňajúci sadu komponentov: <a href="http://tequilla.kmit.sk/~lacike/mingw.zip">http://tequilla.kmit.sk/~lacike/mingw.zip</a>
Subversion	Nástroj na riadenie verzií. Vývojové prostredie vyžaduje len klientsku inštaláciu.	áno	<a href="http://subversion.tigris.org">http://subversion.tigris.org</a>
Trac	Nástroj na riadenie projektu.	nie	<a href="http://trac.edgewall.org">http://trac.edgewall.org</a>

Tabuľka 1: Softvérové systémy tvoriace vývojové prostredie

Nepovinné systémy môžu byť nahradené inými systémami, keďže hlavné nástroje na vývoj sú obsiahnuté v systéme Mingw. Tento dokument avšak iné alternatívy neposkytuje a preto sú v nasledujúcom dokumente uvádzané informácie týkajúce sa tohto referenčného vývojového prostredia.

Vývoj je realizovaný pod operačným systémom Windows. Hoci uvedené nástroje sú dostupné aj pre alternatívne platformy, zdrojové kódy využívajú výlučne funkcionality, ktorá je dostupná pod platformou Windows, preto vývoj na iných platformách momentálne nie je možný.

### 2.2 Zoznam nastavení prostredia

Premenná	Hodnota
Cesta k SVN repositáru	<a href="http://tequilla.kmit.sk/tpsvn">http://tequilla.kmit.sk/tpsvn</a>
Cesta ku adresáru s verziami zdrojových kódov	<a href="http://tequilla.kmit.sk/tpsvn/gos-player">http://tequilla.kmit.sk/tpsvn/gos-player</a>
Cesta k hlavnej vývojovej vetve	<a href="http://tequilla.kmit.sk/tpsvn/gos-player/trunk">http://tequilla.kmit.sk/tpsvn/gos-player/trunk</a>

Tabuľka 2: Zoznam nastavení prostredia

### 3 Inicializácia pracovného prostredia

---

Vývojové prostredie je zložené z nástrojov uvádzaných v tab.1.

#### 3.1 Inštalácia vývojového prostredia

Uvedený postup sa vzťahuje na inštaláciu vývojového prostredia na počítač, kde ešte tieto prostriedky nie sú nainštalované. Pri inštalácii je využívaná štandardná konfigurácia uvedených nástrojov. V prípade, že uvedené prostriedky sú dostupné, nie je potrebné realizovať kroky, ktoré sa týkajú ich inštalácie.

##### 3.1.1 Inštalácia systému Subversion a Mingw:

1. Subversion je možné získať na adrese <http://subversion.tigris.org>. Distribučný balík obsahuje priamo spúšateľné, preto nie je potrebné program inštalovať, ale len rozbaľiť.
2. Adresár v ktorom sa nachádza program `svn` musí byť uvedený v systémovej premennej `PATH`. Nastavenie premennej `PATH` je popísané v sekcii 3.1.3.
3. MinGW je súbor balíkov, ktoré je potrebné postupne rozbaľovať do vybraného adresára. Pre zjednodušenie inštalácia bol vytvorený jeden inštalčný balík. Je dostupný na adrese <http://tequilla.kmit.sk/~lacike/mingw.zip>. Tento balík je potrebné len rozbaľiť.
4. Adresár `bin` v adresári MinGW musí byť uvedený v premennej `PATH`. Nastavenie `PATH` je popísané v sekcii 3.1.3

##### 3.1.2 Inštalácia prostredia Eclipse a potrebných rozšírení

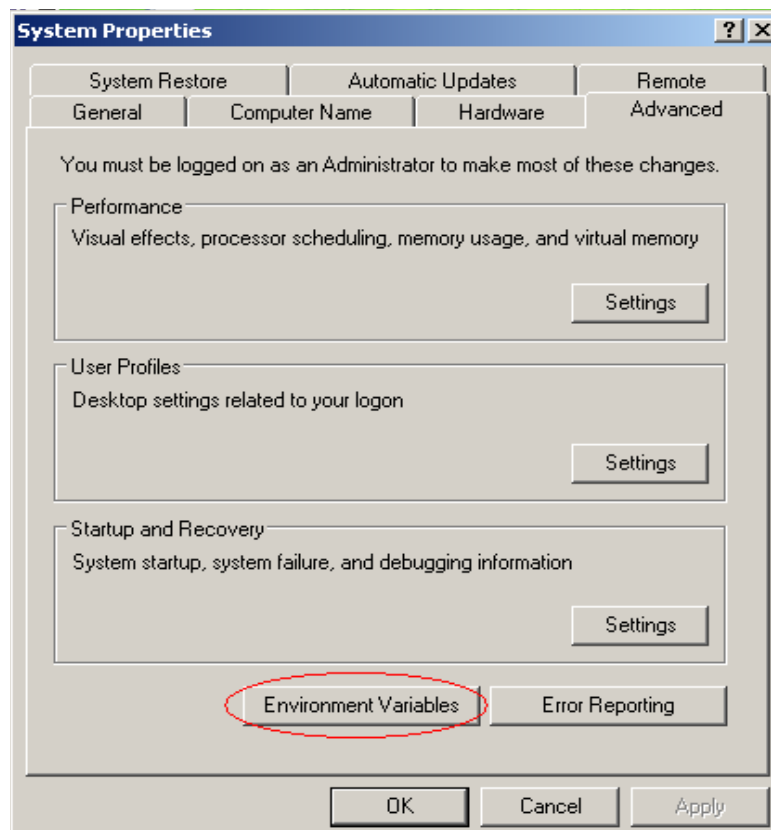
Inštalácia prebieha v nasledujúcich krokoch:

1. Eclipse je možné získať na adrese [www.eclipse.org](http://www.eclipse.org). Balík neobsahuje inštalčné súbory, preto je potrebné ho len rozbaľiť na vybranú cestu.
2. Rozšírenia je možné nainštalovať prostredníctvom dialógu *Find and Install ...*
3. Rozšírenie CDT je dostupné medzi štandardnými rozšíreniami. Rozšírenie Subclipse je dostupné na adrese [http://subclipse.tigris.org/update\\_1.0.x](http://subclipse.tigris.org/update_1.0.x)
4. Po inštalácii rozšírení je potrebné Eclipse reštartovať.

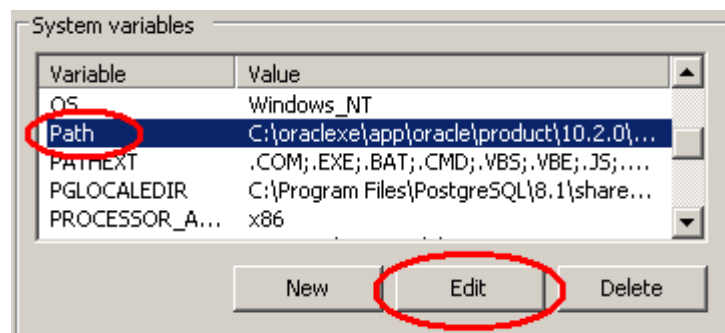
Pri spúšťaní prostredia Eclipse je potrebné určiť adresár, ktorý bude slúžiť ako *workplace*. Lokácia tohto adresára je voliteľná. Nie je potrebné vytvárať ho špeciálnym spôsobom.

##### 3.1.3 Nastavenie premennej PATH

V prostredí Windows XP je nastavenie premennej dostupné prostredníctvom dialógu *System Properties* (obr. 3.1). Dialóg s premennými je zobrazený na obr. 3.2. V premennej `PATH` sú cesty oddeľované bodkočiarkou.



Obrázok 3.1: Prístup k nastaveniu systémových premenných



Obrázok 3.2: Nastavenie systémových premenných

### 3.2 Nastavenie prístupu k SVN a stiahnutie zdrojových kódov

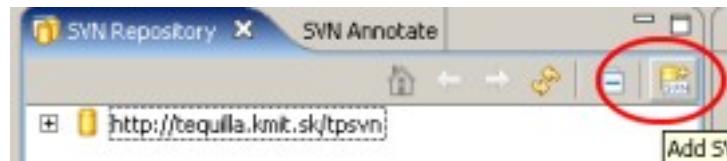
Získanie pracovného prostredia je popisované v prostredí Eclipse. Predpokladom je vytvorený štandardný *workspace*. Odpoča sa použiť samostatný prázdny *workspace*. Je potom možné pracovať s viacerými verziami naraz – viaceré verzie sú reprezentované ako samostatné projekty vychádzajúce z rôznych vetiev v prostredí Subversion.

Nastavenie prístupu k Subversion repositáru:

1. V prostredí Eclipse je nutné zvoliť perspektívu *SVN Repository Exploring*



- Prostredníctvom tlačidla (obr. 3.3) je potrebné spustiť dialóg *Add SVN Repository*

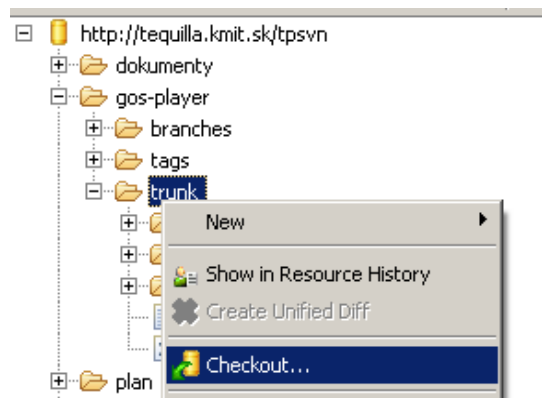


Obrázok 3.3: Tlačidlo pridanie nového repository

- Do dialógu je potrebné zadať cestu k repository. Aktuálna cesta repository je <http://tequilla.kmit.sk/tpsvn>
- Po zadaní cesty k repository je žiadané meno a heslo.
- Po úspešnom zadaní mena a hesla je pridaný repository to panelu *SVN Repository*

Stiahnutie iniciálnej verzie zdrojových kódov:

- V paneli *SVN Repository* je potrebné nájsť vetvu, s ktorou sa má pracovať. V prípade inicializácie pracovného prostredia je predpoklad, že bude využitá hlavná vetva (nastavenie vývojového prostredia – sek. 2.2)
- Po kliknutí na adresár s vetvou pravým tlačidlom je rozvinuté kontextové menu (obr.53.4). V rámci tohto kontextového menu je potrebné zvoliť položku *Checkout ...*



Obrázok 3.4: Voľba pracovnej verzie a získanie vetvy

- Následne je možné špecifikovať meno, pod ktorým bude uvedená vetva uložená do *workspace*. Toto meno nemá vplyv na informácie uložené v repository.
- Po uskutočnení týchto krokov je možné zmeniť perspektívu na *C/C++ Development*. Vo vybranom projekte sa nachádzajú zdrojové súbory.
- V projekte v adresári *src* sa nachádza súbor *Makefile.tmpl*. Tento súbor je potrebné skopírovať do súboru *Makefile*. Na začiatku súboru *Makefile* sú uvádzané premenné, ktoré ukazujú na lib a include adresár systému MinGW. Tieto treba explicitne nastaviť, keďže MinGW sa môže nachádzať v rôznych adresároch.

Po uskutočnení týchto krokov je prostredie pripravené na vývoj.

## 4 Štruktúra pracovného prostredia

Táto časť popisuje štruktúru adresárov v repository, ktoré obsahujú zdrojové kódy a programové výstupy projektu. Štruktúra adresárov, ktoré obsahujú iné artefakty je voľná a preto tu nie je opisovaná.

### 4.1 Adresárová štruktúra projektu

Adresár	Popis
src	Obsahuje zdrojové kódy a hlavičkové súbory programu.
build	Skompilované súbory a súbory s nastavením
ext_include	Tento adresár obsahuje hlavičkové súbory externých knižníc
ext_lib	Tento adresár obsahuje externé knižnice

Tabuľka 3: Adresárová štruktúra projektu

## 5 Správa verzií

---

Táto kapitola opisuje správu verzií ako procesy. Obsahuje opis procesov:

- Vytvorenie novej pracovnej kópie v repozitáry – branching
- Spôsob komentovania zdrojových kódov
- Proces uzatvárania vývoja na pracovnej kópii
- Spájania vetiev

### 5.1 Vytvorenie novej vetvy

Cieľom vytvorenia novej vetvy je vytvorenie izolovaného prostredia. Izolované prostredie je potrebné pre dlhodobý vývoj, keďže nesynchronizované príspevky viacerých vývojárov do jedného hlavného prostredia môžu porušiť konzistenciu.

Účelom novej vetvy je vývoj ucelenej časti, alebo pridávanie novej funkcionality. Na vývojovej vetve by mal pracovať minimálny počet vývojárov – dôvodom je minimalizácia možných konfliktov.

Synchronizácia (spájanie) vývojových vetiev s hlavnou vetvou je realizovaná explicitne (sekcia 5.4).

Vytvorenie novej vetvy je realizované v SVN repozitári. Zmeny sa bezprostredne po vytvorení neprejavajú v lokálnej pracovnej kópii.

Postup:

1. V perspektíve *SVN Repository* je potrebné vybrať zdrojovú vetvu.
2. V kontextovom menu je potrebné zvoliť zobrazenie dialógu *Branch/Tag...*
3. Následne je potrebné špecifikovať cestu k novej vetve. V prípade, že ide o vývojovú vetvu nachádza sa v adresári *branches*.
4. V repozitáry by mal pribudnúť adresár, ktorý obsahuje kópiu zdrojovej vetvy

Následne je možné sa prepnúť na novovytvorenú vetvu. Postup prepnutia na inú vetvu:

1. Predpokladom je perspektíva *C/C++ Development*
2. V kontextovom menu *Team* je potrebné zvoliť položku *Switch*
3. V dialógu je potrebné uviesť cestu k vetve, ktorú chceme používať

### 5.2 Štandardy zápisu komentárov v repozitári

Pri operáciách vytvorenie novej vetvy, ukončovanie vývoja na verve a spájanie vetiev je potrebné rozlíšiť od iných operácií. Pre jednoduchšiu identifikáciu je potrebné používať pri týchto operáciách nasledujúci formát správ.

Operácia	Komentár
Vytvorenie novej vetvy	Branched from <nazov vetvy>
Ukončenie vývoja na vetve	Finished <nazov vetvy> <Popis zmien vo vetve oproti hlavnej vetve, z ktorej táto vetva vychádza>
Spájanie vetiev	Merged <pociatocne cislo revizie>:<konecne cislo revizie> from <nazov vetvy>

Tabuľka 4: Štandardy zápisu komentárov pre riadiace operácie

Tieto komentáre sú záväzné pre uvádzané operácie. Pre bežné operácie – *commit*, a *i.* nie je definovaný špeciálny formát.

Doporučením je po vykonaní zásadnejších zmien napísať jasnú správu, ktorá identifikuje danú zmenu.

### 5.3 Ukončovanie vývoja na vetve

Ukončenie vývoja na vetve je špecifické posledným komentárom pri operácii *commit*. Formát komentáru je uvedený v sekcii 5.2. Následne je potrebné notifikovať človeka zodpovedného za spájanie vetiev – prístup do pracovného repositára je za normálnych okolností obmedzený.

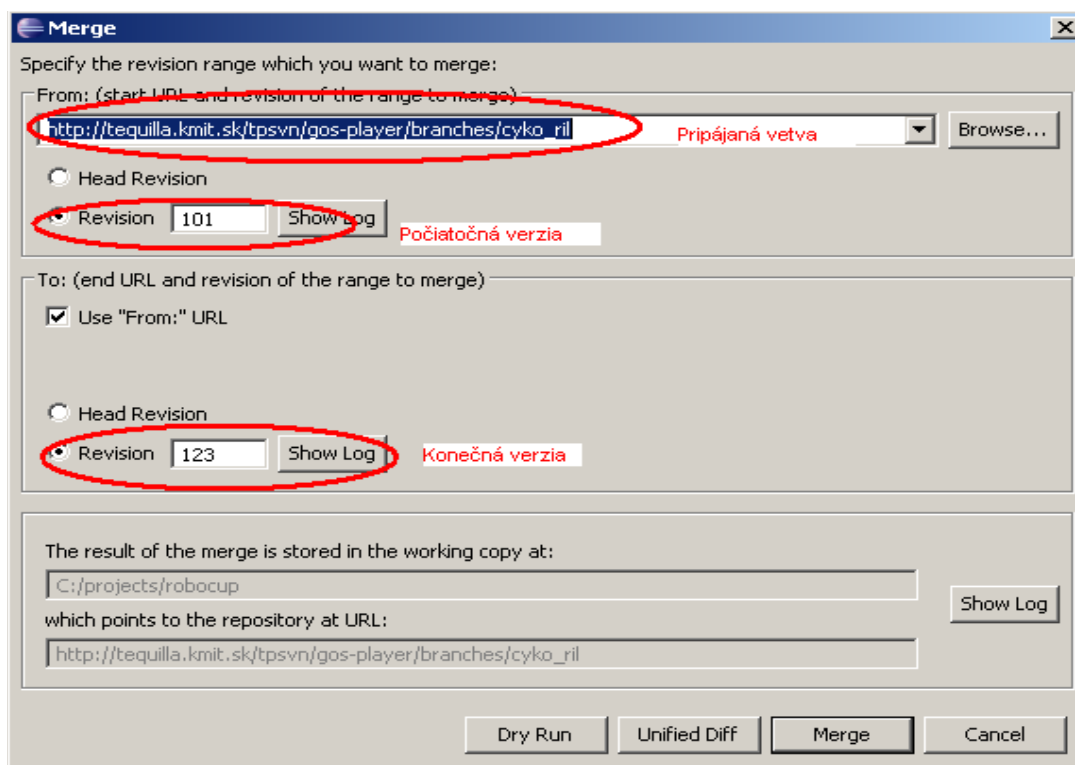
### 5.4 Spájanie vetiev

Cieľom spájania vetiev je združovanie funkcionality, ktorá sa nachádza vo viacerých vetvách. Spájanie vetiev je nutné robiť len na dvoch vetvách naraz. V kontexte systému Subversion je spájanie realizované ako pripojenie vetvy k inej vetve, takže tento proces nie je symetrický.

V prípade, že je nutné spájať viaceré vetvy, je nutné rozdeliť spájanie do viacerých krokov. Tj. Je potrebné iteračným spôsobom k jednej vetve pripájať ďalšie. Po každom spojení dvoch vetiev je potrebné vykonať operáciu *commit*. Spájanie je potrebné označiť komentárom (sekcia 5.2)

Postup na spájanie vetiev (prostredie Eclipse/Subclipse):

1. Predpokladom je že pracovná kópia vo *workspace* je získaná z vetvy, ku ktorej sa má byť realizované pripojenie. Tj. ak má byť do vetvy *trunk* pripojená nejaká iná vetva, tak vo *workspace* musí byť vetva *trunk*.
2. Je potrebné získať informácie o číslach verzií v SVN, ktoré korešpondujú so začatím vývoja na vetve (číslo verzie, keď bola vetva vytvorená) a koncom vývoja na vetve. Tieto informácie je možné získať prezeraním histórie verzií v SVN. Je možné využiť prostredie Eclipse, ale aj iný nástroj. Tieto verzie je možné nájsť podľa komentárov (sekcia 5.2)
3. Samotné pripájanie je realizované pomocou dialógu *Merge...*. Tento dialóg je dostupný prostredníctvom kontextového menu *Team*.
4. V dialógu *Merge* je potrebné uviesť informácie o počiatkovej a konečnej verzii a vetvu, ktorá sa má pripájať.



Obrázok 5.1: Dialóg Merge

5. Po ukončení tohto kroku je do pracovnej verzie pripojená vetva. **Pripojenie je len lokálne, zmeny ešte nie sú uložené v repositári.**
6. Pri pripájaní môžu vzniknúť konflikty. Tieto je potrebné vyriešiť ručne. Prostredie Eclipse zvýrazňuje konflikty.
7. Po vyriešení konfliktov je nutné realizovať operáciu *commit*. Cieľová vetva závisí od účelu spájania. Vetva, ktorá bola pripojená sa už nesmie používať. V prípade, že ide o vývojovú vetvu, odporúča sa ju zmazať. Výsledná vetva je uložená ako označená verzia, alebo hlavná vetva.

V prípade spájania do hlavnej vetvy, alebo pri tvorbe označených vetiev (adresár *tags*) je potrebné viesť evidenciu o funkcionalite v danej verzii. K hlavnej vetve a k označeným vetvám je preto nutné viesť evidenciu o funkcionalite, ktorá sa v nich nachádza. Táto evidencia zadávaná do systému Trac.

## 6 Štandardy písania kódu

Pri písaní kódu sú vopred dohodnuté konvencie. Existujúci kód bol upravený tak, aby ich spĺňal. Tvorené sú nasledovnými pravidlami.

### 6.1 Vizuálna stránka kódu

- Šírka riadku je maximálne 80 znakov, potom sa zalamuje tak aby boli logicky zarovnané s riadkom (napravo priradenia =, vstupné premenné volanej funkcie zarovnané pod seba)
  - Oddeľovač blokov sú 4 medzery, teda žiaden tabulátor. Začiatok { a koniec } bloku je vždy na samostatnom riadku a patria pod prvý znak predošlého riadku príkazové bloky ktoré majú iba jeden príkaz sú bez { a }.
  - Všetky písmena sú veľké, slová sú oddelené \_ (podtrhovníkom) (príklad: MAX\_PLAYER\_TYPES)
  - Každé slovo v názve tried a typov má prvé písmeno veľké (príklad: WorldManager?), slová sú spojené bez špeciálnych znakov a sú usporiadané podľa dôležitosti (alebo rozšírenia informácie) zľava doprava (príklady: Player, PlayerType, PlayerTypeInfo)
  - Každé slovo v názve metód a funkcií má prvé písmeno veľké okrem prvého slova (príklad: setPlayerType), slová sú spojené bez špeciálnych znakov a sú usporiadané podľa dôležitosti (alebo rozšírenia informácie) zľava doprava. Pri metódach settery majú prvé slovo set, pri metódach gettery majú prvé slovo get.
  - Pri definícii vstupných premenných každá premenná musí byť na novom riadku zarovnaná za názov funkcie, pričom názvy premenných musia byť zarovnané pod seba
  - Deklarovanie metód a funkcií (platí aj pre if, switch, while, atď.) - začiatok označenia vstupných premenných ide bezprostredne za názov funkciu (príklad: setType( )). Ak nie sú vstupné premenné, tak sa medzi zátvorky dáva jedna medzera. Ak sú vstupné premenné, tak sa za prvú zátvorku dáva medzera a pred poslednú zátvorku dáva medzera (príklad: setPosition( double x, double y )). Vstupné premenné sa oddeľujú čiarkou, ktorá nasleduje bezprostredne za definíciou premennej, za čiarkov je jedna medzera
  - Názvy premenných sú obdobné ako názvy metód / funkcií, ale prvé slovo je skratka typu
    - d - double
    - i - interger
    - s - char\*
    - c - char
    - ak sa jedná o iný typ alebo triedu tak sú vybrané a spojené všetky veľké písmena z názvu triedy
- Prípadne sa ako slová použijú skratky maximálne 2-4 písmen opisujúcich akú informáciu premenná obsahuje (príklad: Vector vecPos).
- Pred a za každý binárny operátor sa dáva medzera (teda operátory =, ==, &, &&, |, , +, -, \*, /, <<, >>, <, >, atď.). Za unárny operátor sa medzera nedáva (teda operátory ako !, ++, --, referenčný operátor &, operátor ukazovateľa \*). Pre operátor výberu z poľa [ ] platia rovnaké podmienky ako pre zátvorky pri volaní metód / funkcií operátor dereferenčného volania metódy ->, definície metódy/statického volania :: alebo priameho volania . je bez

medzier ako vľavo a tak aj vpravo

- Každý riadok sa začína // komentár typu /\* sa používa iba pri komentároch určených na tvorbu dokumentácie pomocou nástroja Doxygen. Ak sa dá komentáre by mali byť zarovnané pod seba .

## 6.2 Pravidlá pre použitý jazyk

- Na vstup a výstup sa používajú streamy. Štandardný výstup je definovaný ako premenná (štandardne cout). Štandardný chybový výstup je cerr.
- Každý hlavičkový súbor sa začína riadkami

```
#ifndef _NÁZOV_  
#define _NÁZOV_
```

kde `_NÁZOV_` je názov súboru bez prípony a všetky písmena sú veľkým

- Používa sa dátový typ double a nie float.
- Pri vracaní cez return sa daná výstupná premenná/hodnota nikdy neuzatvára do zátvoriek.

## Prílohy

---

### **A. Bibliografia**

### **B. Zoznam tabuliek**

Tabuľka 1: Softvérové systémy tvoriace vývojové prostredie.....	II
Tabuľka 2: Štandardy zápisu komentárov pre riadiace operácie.....	VIII

### **C. Zoznam obrázkov**

Obrázok 3.1: Prístup k nastaveniu systémových premenných.....	4
Obrázok 3.2: Nastavenie systémových premenných.....	4
Obrázok 3.3: Tlačidlo pridanie nového repository.....	5
Obrázok 3.4: Voľba pracovnej verzie a získanie vetvy.....	5