

1 ÚVOD

Tento dokument obsahuje popis pravidiel, ktoré sa budú dodržiavať pri písaní zdrojových textov futbalového hráča.

Existujúce zdrojové texty na niektorých miestach nie sú komentované dostatočne. Tieto komentáre nebudeme dopĺňať, len ak na to bude čas v závere. Ku zdrojovému kódu sa budú vytvárať komentáre presne podľa týchto pravidiel. Komentáre sa budú vkladať do textu aj vtedy, ak by to znamenalo zdržanie práce na zdrojových textoch, pretože „neskôr“ ich už do zdrojového kódu nikto dopĺňať nebude a program by sa stal neprehľadným.

2 IDENTIFIKÁTORY TRIED, FUNKCIÍ A PREMENNÝCH

Názov štýlu	Príklad
Pascal	SmallRedPoint
Camel	smallRedPoint
C style	small_red_point
Hungary	mSmallRedPoint, m_SmallRedPoint, intSmallRedPoint
Basic	SMALL_RED_POINT

2-1 Štýly zápisu identifikátorov

Popis štýlov použitých na identifikátory premenných je špecifickejšie popísaný pri každom použití v ďalších kapitolách.

Identifikátory lokálnych premenných sa budú zapisovať výhradne pomocou štýlu *Camel*. Takýmto istým štýlom sa budú zapisovať aj identifikátory parametrov funkcií.

Na identifikátory premenných sa budú používať obsahovo významné slová a slovné spojenia, nie generické názvy. Obsahovo nevýznamné slová, prípadne len jednotlivé písmená je možné použiť len na lokálne pomocné premenné, prípadne na premenné v iterátoroch cyklov. Skratky sa budú používať ako identifikátory len v nevyhnutných prípadoch a len vtedy, keď je možné ľahko zistiť ich význam.

Každá premenná bude deklarovaná na samostatnom riadku. Za každou premennou bude uvedený komentár, na čo slúži.

2.1 Trieda

2.1.1 Identifikátor

Používa sa *Paskalovský* štýl bez predpony. Pred názov triedy sa nevkladá predpona C.

2.1.2 Členské premenné

V zdrojových textoch tímu Sklo sa nachádzajú triedy so štýlom *Camel* a *Hungary* (použitá predpona „m_“). Nerozlišuje sa pri používaní štýlu medzi jednotlivými úrovňami viditeľnosti členskej premennej.

Existujúce identifikátory sa prepisovať v zdrojových kódoch nebudú. My budeme vo svojom kóde používať štýl *Hungary* s predponou „m_“.

2.1.3 Členské funkcie

Na identifikátory členských funkcií tried sa bude používať výhradne *Paskalovský* štýl zápisu.

2.2 Štruktúra

Štruktúry budú pre nás predstavovať len zhromaždisko údajov. Preto sa v nej nebudú vyskytovať žiadne členské funkcie. V prípade, že je treba nad danými údajmi vytvoriť nejakú funkciu, tak sa zabalí do triedy nie do štruktúry.

Na identifikátory sa používa štýl mnou nazvaný ako *Basic*. Jednotlivé premenné v štruktúre sa zapisujú pomocou štýlu jazyka C, alebo pomocou štýlu *Camel*. Odporúča sa použiť štýl *Camel*.

2.3 Vymenovaný typ

Identifikátor aj všetky hodnoty vo vymenovanom type sa budú zapisovať výhradne *Paskalovským* štýlom.

3 KOMENTÁRE

3.1 Dokumentačné komentáre

Pre podporu dokumentovania zdrojových súborov sa bude používať systém Doxygen. Dokumentácia pre Doxygen začína ako obyčajný komentár pre C++. Obsahuje však jednu lomku, alebo hviezdičku navyše:

```
/// Jednoriadkový (stručný) komentár  
/** Viacriadkový (úplný) komentár  
* /
```

Stručný komentár sa môže nachádzať v dokumentačnom bloku spolu s úplným komentárom. V tomto prípade ho vyznačíme slovom `\brief`.

```
/** \brief Skrateny komentar,  
* v tomto pripade prehadza viaceremi riadkami.  
*  
* od uplneho komentara bol oddeleny prazdnym riadkom  
*/
```

Doxygen poskytuje kľúčové slová pre zápis významu informácií uvedených v komentároch. Najzaujímavejšie sú tieto:

```
\brief strucny popis v uplnom popise  
\author Meno autora  
\param menoArgumentuFunkcie Popis vyznamu argumentu funkcie  
\returns Popis vyznamu navratovej hodnoty funkcie  
\todo Popis veci, ktore zamyslame dorobit  
\file Popis obsahu súboru
```

3.2 Komentáre jednotlivých entít zdrojového kódu

Každý jeden súbor, ktorý vytvoríme, bude mať na začiatku uvedenú hlavičku. Táto bude obsahovať názov súboru, stručný popis daného modulu, ktorý je obsiahnutý v súbore. Bude sa v ňom nachádzať tiež meno autora a dátum vytvorenia súboru.

```
/**
 * \file Priklad.h
 * Interfejs pre triedu Priklad
 *
 * \author Jozko Mrkvicka
 * \date 23.11.2002
 */
```

Prípadné definície makier budú okomentované ako obyčajná premenná, ak sa jedná o makro bez parametrov. Ak sa jedná o makro s parametrami, tak sa bude komentovať ako funkcia.

Každá použitá premenná bude pri svojej deklarácii okomentovaná stručným jednoriadkovým komentárom. Toto platí pre členské premenné tried, štruktúr aj vymenovaných typov, pre lokálne premenné a aj pre globálne premenné. Členské premenné tried, štruktúr a vymenovaných typov, ktoré sú viditeľné zvonka (public a protected) a tiež globálne premenné budeme komentovať stručným dokumentačným komentárom. Lokálne premenné funkcií a premenné s modifikátorom private budú komentované klasickým komentárom.

Každá funkcia bude pri deklarácii v triede okomentovaná stručným jednoriadkovým komentárom, ktorý bude popisovať jej hlavnú funkcionálnosť. Pri svojej definícii bude funkcia okomentovaná podrobne viacriadkovým komentárom. V tomto komentári sa bude nachádzať stručný popis toho, čo funkcia robí, popis jednotlivých parametrov, popis návratovej hodnoty (ak nie je typu void). Ak bude funkcia generovať výnimku (predpokladám, že toto nikto z nás nebude používať, lebo to veľmi spomaľuje real time aplikácie), tak sa výnimka, ktorú bude generovať tiež uvedie v komentári funkcie. Taktiež bude v komentári ku funkcii uvedený jej autor.

```
/**
 * Parsuje spravy zo servera verzie 10.0.1, ktore
 * su zapisane v jazyku CLang 7.0. Spravy uklada do
 * vystupnej fronty spracovanych sprav.
 *
 * \author Jozko Mrkvicka
 *
 * \param buf Retazec, ktory prisiel zo servera.
 * \param out Front spracovanych sprav.
 * \returns 0 ak spracovanie prebehlo uspesne, -1 v pripade
chyby.
 */
int ParseCoach( char *buf, Fifo &out ){ ... }
```

Každá definícia triedy, štruktúry a vymenovaného typu bude obsahovať viaciadkový komentár. V ňom sa bude nachádzať jej popis a meno autora.

```
/**
 * \author Jozko Mrkvicka
 * \brief Strucny popis triedy.
 *
 * Alebo uplny popis triedy.
 * Moze mat viacero riadkov.
 */
```