Dávid Lázár 1.3.2016

Dokumentácia k zápočtovému programu

# Úvod

Zadaním zápočtového programu je prepísať zdrojový kód v Pascale do súboru, ktorý sa dá zobraziť vo webovom prehliadači. Program farebne odlišuje slová a znaky a formátuje bloky textu – popísané v TabulkaDokumentacie. Ak je v jednom riadku viac príkazov s bodkočiarkami za príkazmi (prípadne viacero komentárov k príkazom za príkazmi), tak ich program rozdelí pod seba do riadkov a odsadí – popísané v príkladoch.

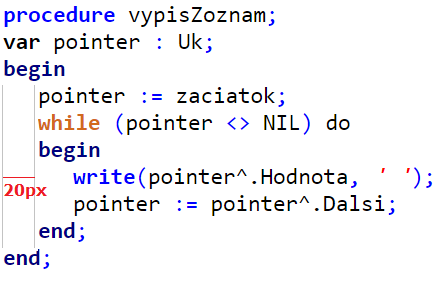
# Užívateľská časť

Program ako vstup načíta textový súbor vstup.txt, v ktorom má byť syntakticky správne zapísaný zdrojový kód v Pascale. Zo vstupu program prepíše index.html súbor, v ktorom sa bude nachádzať zvýraznený kód. Oba súbory sa musia nachádzať v rovnakom priečinku ako program.

Na zobrazovanie kódu som použil tag [<code>](http://www.w3schools.com/tags/tag_code.asp), ktorý podľa w3c definuje počítačový kód.

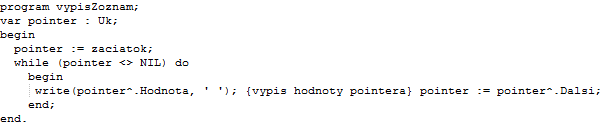
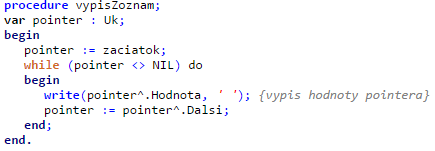
Zvýraznenie a formátovanie je popísané v prílohe - excelovskom dokumente TabulkaDokumentacie.xlsx.

Po begin, repeat, until, end a case sa mení odsadenie. Odsadením sa vo výstupe myslí to, že sa text od begin, case alebo repeat až po jeho párový end alebo until blokovo odsadí o 20px zľava. Počet pixelov, o koľko sa má text odsadiť sa dá zmeniť v súbore styles.css.



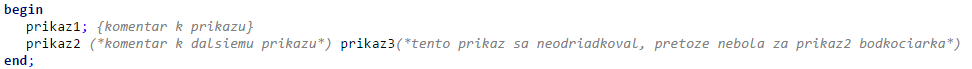
Na obrázku je názorne zvýraznené odsadenie textu. Program žiadne sivé ani červené čiary pri odsadzovaní nekreslí.

Ak je vo vstupnom zdrojovom kóde viac príkazov v jednom riadku, tak sa vo výstupe odriadkujú a zarovnajú pod seba. Ak sa nachádza komentár za príkazom, tak ho program chápe ako komentár k príkazu v riadku, preto odriadkuje až príkazy za ním.

Pre príklad:   
zo vstup.txt :   
  
sa vygeneruje index.html, ktorý sa zobrazí vo webovom prehliadači: 

zo vstupu:

Program vygeneruje



Komentár typu (\* \*) sa vypíše rovnakým spôsobom ako komentár typu { }.

Pre zmenu štýlu slúži styles.css.

Slová, ktoré program dokáže rozlíšiť sa dajú pridávať jednoduchým a rýchlym upravením zdrojového kódu.

Ostatné slová program navzájom nijako neodlišuje.

Program bol ladený na operačnom systéme Windows 8, ktorý ukladá konce súborov ako kombináciu znakov CR LF. Program je prispôsobený Windows-u (odskúšané), mal by fungovať na Linuxe a nefunguje na Mac (neodskúšané).

# Programátorská časť

Program je zostavený z jedného hlavného cyklu, ktorý sa opakuje, kým existuje znak, ktorý by mohol program načítať. Cyklus načítava znaky po jednom, a následne sa rozhoduje, čo s ním na základe predchádzajúcich znakov spraví.

Pokiaľ je načítaný znak písmeno, bodka, apostrof alebo podčiarkovník, tak sa buď vytvorí nové slovo alebo sa pridá načítaný znak k slovu, ktoré sa načítavalo z predchádzajúcich znakov. Ak načítaný znak nie je nijak iný ako písmeno, bodka, apostrof, podčiarkovník, tak sa slovo vypíše, prípadne zvýrazní a vypíše.

Po dokončení cyklu sa musí ešte vypísať posledné slovo ’end.’. Nie je zložité nechať toto slovo vypísať ešte cyklom, ale to by pridalo ďalšiu podmienku na vyhodnotenie pre každé prejdenie cyklom. Efektívnejšie je ušetriť podmienku z cyklu a radšej nechať vypísať slovo na konci príkazom.

V programe sú použité premenné **index**, **vstup** – na otváranie vstupného súboru vstup.txt na čítanie a index.html na prepísanie.   
Ďalšie premenné znak **typu** char a **slovo** typu string slúžia v programe na načítavanie znakov a následne skladanie znakov do slov pre výpis.   
Booleanské premenné použité v programe sú:   
**bolDiv**, **specDiv** – využité pre správne odsadenie textu. **bolDiv** a **specDiv** sa líšia v tom, kedy program začne odsadzovať text.   
**bolaBodkociarka**, **priznakBodkociarkyPredKomentarom** – využité pre odriadkovanie viacerých príkazov za sebou v jednom riadku.   
**zatvorkovyKomentar**, **mnozinovyKomentar**, **otvaraciaZatvorka**, **zatvaraciaHviezdicka** – využité pri rozpoznávaní začiatku a konca komentárov.  
**retazec** – využitá pri rozpoznávaní začiatku a konca reťazca.  
Tieto premenné budú ešte spomenuté nižšie.

Procedúra **zapisZaciatokHTML** – zapíše otvárací tag HTML dokumentu, hlavičku HTML, a zapíše otvorenie tela HTML. Vypíše sa aj tag <code>, ktorý značí začiatok nejakého kódu zobrazovaného na webe podľa w3c (link vyššie).

Procedúra **zapisKoniecHTML** – zapíše zatvárací tag </code>, zatvárací tag tela HTML a zatvárací tag HTML dokumentu.

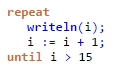
Procedúra **pridajZnakSlovu** – načítaný znak v premennej typu char **znak** pripíše k stringu **slovo**, ktoré bolo načítané z predchádzajúcich znakov. V prípade, že slovo je prázdne, tak sa slovom stane načítaným slovom.

Program na vypisovanie nového riadka používa dva spôsoby:   
1) nepárovým tagom <br>   
2) pomocou <div class=“posun“> a </div>, kde posun je identifikácia, ktorá podľa styles.css odsadí text v novom riadku o toľko pixelov, koľko je zapísané v styles.css.

Procedúra **zapisEnter** – úlohou procedúry je správne zapísať nový riadok.   
V prípade **specDiv** = True program vie, má ako nový riadok použiť blokové odsadenie <div class=“posun“>. Príznak **specDiv** sa nastavil na hodnotu True napríklad pri načítaní slova case ( za slovom case sa ešte očakávajú nejaké premenné a slová).   
V prípade, že bolDiv je False program vie, že sa má použiť odriadkovanie tagom <br>.

Procedúra **vypisSlovo** – úlohou procedúry je vypisovať slová.   
Ak je slovo súčasťou komentára alebo reťazca, tak sa vypíše bežným príkazom – o formátovanie textu sa postarajú procedúry na rozoznanie komentárov a reťazcov.   
Inak ak je **slovo** neprázdne – teda ho tvorí aspoň jeden znak, tak sa podľa vetvenia zistí o aké slovo sa jedná a aplikuje sa štýl zo styles.css.   
Ak je **slovo** begin, tak sa slovo vypíše formátom z TabulkaDokumntacie.xlsx a nastaví sa **bolDiv** na hodnotu true. Keďže <div class=“posun“> už zapíše nový riadok, musí sa koniec riadku (pri načítaní) spracovať a nezapísať ďalší nový riadok. To je ošetrené nastavením príznaku **bolDiv** na hodnotu True a následným vetvením pri volaní procedúry **zapisEnter**.   
Ak je slovo end alebo end., tak sa slovo vypíše formátom z TabulkaDokumntacie.xlsx a zapíše zatvárací tag </div>, ktorým program uzavrie blok textu (napríklad medzi begin a end;). Nastaví sa **bolDiv** na false, aby ďalšie konce riadkov boli spracované a vypisované tagom <br>.  
Ak je slovo case, tak sa slovo vypíše formátom z TabulkaDokumntacie.xlsx. Ďalej sa nastavia oba príznaky **bolDiv** a **specDiv** na hodnotu True, pretože program ešte očakáva nejaký text medzi koncom slova a koncom riadku. Tento text program schválne neodriakuje, pretože ho očakáva v tvare:  
premenná of – program za týmto textom očakáva už len koniec riadku (funguje bez konkrétnych úprav v kóde).   
Ak je slovo repeat, tak sa slovo vypíše formátom z TabulkaDokumntacie.xlsx a nastaví sa príznak **bolDiv** na hodnotu True - aby sa text za repeat (až po until) na mohol blokovo oddeliť a odriadkovať.  
Ak je slovo until, tak sa zapíše zatvárací tag </div> a slovo sa vypíše formátom z TabulkaDokumntacie.xlsx. Zaznačí sa, že sa zapísal </div>, teda že program ďalej zapisuje riadky pomocou tagu <br>.  
Pre ostatné slová, ktoré sú v TabulkaDokumentacie.xlxs sa vypíše **slovo** formátom popísaným v dokumente TabulkaDokumentacie.xlxs.   
Pre tie slová, ktoré nie sú v dokumente TabulkaDokumentacie.xlxs platí, že sa vypíšu tagom <output>slovo</output>.   
Špeciálnym prípadom sú slová begin a repeat, pretože za týmito dvoma slovami môže nasledovať niekoľko príkazov a slovo end alebo until . Program ich má za úlohu odriadkovať a odsadiť zľava. Kvôli tomu sa nastaví príznak **bolaBodkociarka** na True. Tým program imituje načítanie bodkočiarky a odriadkuje nasledujúci text ako keby za slovami begin alebo repeat bola bodkočiarka.  
Pre príklad:

  
Vygeneruje index.html, ktorý sa zobrazí:  


  
Vygeneruje index.html, ktorý sa zobrazí:  
  
Po vypísaní obsahu premennej **slovo** sa vynuluje.

Keďže som program ladil na operačnom systéme Windows (ktorý značí koniec riadku ako kombináciu znakov s ASCII hodnotami 10 a 13 (CR a LF)), tak pre uľahčenie ladenia a fungovania programu, program vypisuje nový riadok pri znaku LF a znak CR nechápe ako koniec riadku.

Procedúra **posudLF** je volaná, ak je načítaný znak LF. Procedúra zavolá **zapisEnter** a pokiaľ bol príznak **bolaBodkociarka** = True, tak sa hodnota zmení na false. Fungovanie a spracovanie bodkočiarky je popísané nižšie.

Procedúra **zapisRetazec** je volaná, ak je načítaný znak apostrof (ktorý v Pascale značí začiatok alebo koniec reťazca).  
 Ak bol apostrof načítaný v komentári, tak sa len vypíše (o formátovanie komentára sa postarajú procedúry na otváranie a zatváranie komentárov).   
Ak bol apostrof načítaný mimo komentára a zároveň mimo reťazca tak apostrof program chápe ako začiatok reťazca a teda aplikuje štýl z TabulkaDokmuntacie tagom <output class=“retazec“> a vypíše znak, ďalej sa nastaví príznak **retazec** na hodnotu true.   
Ak bol apostrof načítaný a zároveň je príznak **retazec** nastavený na hodnotu true, tak apostrof program chápe ako ukončenie reťazca a zapíše ukončovací tag znak a </output>, ďalej sa nastaví príznak **retazec** na hodnotu false.

Procedúra **zaciatokMnozinovehoKomentaru** je volaná ak je načítaný znak otváracia množinová zátvorka. Ak načítaný znak nie je súčasťou komentára alebo reťazca, tak sa zátvorka vypíše s formátovaním z TabulkaDokumentacie.xlsx a nastaví sa príznak **mnozinovyKomentar** na hodnotu True. Inak sa zátvorka len vypíše – o formátovanie množinovej zátvorky v reťazci sa postarajú procedúry **zapisRetazec**.

Procedúra **koniecMnozinovehoKomentaru** je volaná, ak je načítaný znak zatváracia množinová zátvorka. Prvým krokom procedúry je výpis znaku. Ak je počas načítania znaku zároveň znak mimo reťazca a je príznak **mnozinovyKomentar** na hodnotu true a príznak **zatvorkovyKomentar** na hodnotu false, tak sa množinový komentár znakom ukončí, to znamená, že príznak **mnozinovyKomentar** sa nastaví na hodnotu false a do výstupu sa zapíše ukončenie komentárového textu tagom </output>.

Procedúra **posudOtvaraciuZatvorku** je volaná, ak je načítaný znak otváracia zátvorka (. Procedúra rozlišuje ako vypísať otváraciu zátvorku (. Ak je súčasťou reťazca alebo ktoréhokoľvek komentára, tak sa jednoducho vypíše. Inak sa nastaví príznak **otvaraciaZatvorka**. Tento príznak sa neskôr použije pri posudzovaní znaku hviezdičky – pričom kombinácia (\* otvára v Pascale kód.

Procedúra **posudHviezdicku** je volaná ak je načítaný znak \*. Táto procedúra je trocha zložitejšia ako ostatné, pretože rozhoduje o štyroch možnostiach: otvorenie zátvorkového komentára, možné ukončenie zátvorkového komentára, výpis v komentári alebo reťazci (teda prenechanie formátovania na iné procedúry) alebo výpis hviezdičky ako znamienka – procedúrou **zaznacZnamienko** (popísaná nižšie).  
Ak je \* v reťazci, tak sa vypíše bez formátovania procedúrou **posudHviezdicku**, to isté sa spraví v prípade, že v množinovom komentári.   
Ak hviezdička nie je ani v komentári ani v reťazci, tak ju program chápe ako jedno zo znamienok a zavolá procedúru **zaznacZnamienko**, ktorá sa postará o formátovanie \* ako znamienka.  
V prípade, že sa hviezdička načíta hneď po otváracej zátvorke, tak sa zapíše formátovanie komentára, vypíšu sa znaky ( a \* za sebou a nastavia sa hodnoty True do príznaku **zatvorkovyKomentar** a false do **otvaraciaZatvorka**.   
Inak, ak sa hviezdička nachádza v komentári, tak môže nastať prípad, že po hviezdičke sa zo vstupu načíta zatváracia zátvorka ) a bude sa môcť zátvorkový komentár uzavrieť – v takom prípade sa nastaví príznak **zatvaraciaHviezdicka** na hodnotu True.

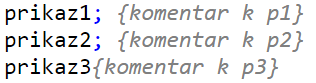
Procedúra **posudZatvaraciuZatvorku** je volaná, ak je načítaný znak ). Ak nie je ) súčasťou komentára ani reťazca, tak znak ) procedúra **zaznacZnamienko** vypíše s formátovaním znamienka z TabulkaDokumentacie.xlsx. Inak ak je zatváracia zátvorka ) vnútri zátvorkového komentára a zároveň platí, že predchádzajúci znak bola \* (stačí sa pýtať na príznak **zatvaraciaHviezdicka**, keďže v programe sa dá dosiahnuť **zatvaraciaHviezdicka** = True len ak je znak načítaný v zátvorkovom komentári – teda **zatvorkovyKomentar** = True), tak program nastaví hodnoty false do oboch príznakov **zatvaraciaHviezdicka** a **zatvorkovyKomentar** a vypíše zatvárací tag </output>, ktorým ukončí formátovanie komentára.

Procedúra **posudZnamienko** je volaná ak je načítaný znak jeden zo znakov + , - / : < = > [ ] ` alebo procedúrami **posudZatvaraciuZatvorku**, **posudHviezdicku** a **posudBodkociarku**. Ak je načítaný znak vnútri komentáru alebo reťazca, tak sa vypíše bez formátovania – o formátovanie sa postarajú procedúry **zapisRetazec** a procedúry, ktoré otvárajú alebo zatvárajú komentáre, prípadne procedúry, ktoré rozhodujú o znamienkach \*, ).

Procedúra **posudBodkociarku** je volaná ak je načítaný znak ;. Pokiaľ sa znak nenachádza v komentári alebo reťazci, tak ho program vypíše bez formátovania - – o formátovanie sa postarajú procedúry **zapisRetazec** a procedúry, ktoré otvárajú alebo zatvárajú komentáre. Inak sa nastaví príznak **bolaBodkociarka** = True a zaznačí sa ako znamienko procedúrou **zaznacZnamienko**.

Hlavný program začína otvorením súborov na čítanie a zápis, následne sa dosadia hodnoty false do príznakov a zapíše sa začiatok HTML do index.html. V hlavnom cykle programu sa pred rozoznávaním znaku musia vykonať 3 kontrolné vetvení:  
1. prepísanie príznaku **otvaraciaZatvorka** na hodnotu False a výpis znamienka (, ak po znaku ( nenasleduje \*,  
2. prepísanie príznaku **zatvaraciaHviezdicka** na hodnotu False ak po znaku \* nenasleduje ),  
3. vypísanie slova, pokiaľ znak nie je znakom, ktorý program chápe ako znak slova – tie sú: . ^ \_ alebo jedno z malých alebo veľkých písmen.

Aby program dokázal odriadkovať do riadkov viaceré príkazy z jedného riadku, má program v hlavnom cykle v prípadoch znak = (, ), ;, znamienka, znaky slova, {, } pridané ošetrenia. Hlavnou myšlienkou týchto ošetrení je zabezpečenie súdržnosti komentárov s príkazmi.  
Pre príklad:  
prikaz1; {komentar k p1} prikaz2; {komentar k p2} prikaz3{komentar k p3} program vypíše vo webovom prehliadači.



Ďalšie príklady sú v užívateľskej časti.  
Na konci hlavného cyklu program načíta nový znak a pokračuje ďalej.   
Program sa končí výpisom posledného slova, dopísaním konca HTML dokumentu a uzavretím súborov.