Šeste vaje iz računalništva I

Alen Kramar

17. 11. 1998

Povzetek

Na teh vajah bomo izdelali dva podprograma, ki ju bomo vključili v `KVADRAT.C`. Uporabili bomo stavka `WHILE` in `FOR`, nato pa bomo naš kvadrat spremenili v pravokotnik in zapolnili z zvezdicami.

Izdelava podprograma za izris osnovnice in vmesnice v programu kvadrat.c

Podprogram za izris osnovnice

Izrisali bomo osnovnico, ki bo široka, kot željena širina.

Najprej se moramo vprašati *kaj* bomo naredili in potem *kako* bomo to naredili. Po premisleku pridemo do spoznanja, da bomo to izvedli s pomočjo funkcije, ki jo bomo poimenovali `izris_osnovnice`. Ta funkcija nam bo izpisovala „+“ in „-“ – slednje bomo izvedli z stavkom *for*. Naš stavek `for` nam bo risal „-“ tako dolgo, dokler bo vrednost števca `i` (začetna vrednost `i = 1`) ustrezala pogoju $\mathtt{i} \leq \mathtt{sirina} - 2$. Ko ta pogoj ne bo več izpolnjen, pa se bo izpisal še zadnji „+“.

Stavek `for`

V tem podprogramu smo uporabili stavek `for`, ki je sestavljen na sledeč način:

`for(izraz1;izraz2;izraz3)stavek;`

Kaj pa to pomeni?

1. izračunaj `izraz1`;
2. če je pogoj, kot ga določa `izraz2` izpolnjen (resničen), ...
3. ... potem izvedi `stavek`;
4. izračunaj `izraz3` in se vrni na 2.

Podrprogram za izris vmesnice

Sedaj moramo izdelati še funkcijo `izrisi_vmesnice`. Ta funkcija je napisana na podoben način kot `izrisi_osnovnico`, samo da v stavku `for` ne bomo izpisoval „-“ in „+“ ampak „ “ in „—“.

Sedaj smo naredili domačo nalogo.

Zanka `do ... while`

Zanko `do-while` uporablajmo tam, kjer imamo določen stavek, ki naj bi se izvajal toliko časa, dokler bo vrednost izraza izpolnjena (resnična).

Zanko `do-while` bomo vrinili v glavni program (`main`), da nam bo program izrisal naš poljuben kvadrat večkrat ne da bi ponovno zaganjali program.

V glavnem programu, definiramo novo spremenljivko / kosntanto *najmanjša širina* – `najmanjsa_sirina`. To storimo na sledeči način:

- definiramo jo moramo na začetku programa, da je takoj vidna;
- in `const int najmanjsa_sirina = 2`, kar pomeni, da je `najmanjsa_sirina` konstantno (v prograu se ne spreminja število, ki ima vrednost 2.

Tako definirano konstanto uporabimo na dveh mestih v programu:

- v izstopnem pogoju `do-while` zanke v glavni funkciji in
- pri izpisu navodil uporabniku, kaj naj vpiše.

Zaradi definicije konstante sedaj samo na enem mestu lahko popravimo najmanjšo širino, sprememba pa se bo odražala na obeh mestih. To pomeni lažje vzdrževanje programa.

Popravljanje oziroma krajšanje programa

Naš kvadrat bomo zapolnili z „`*`“ ter funkciji za izris osnovnice in vmesnice malo popravili, oziroma skrajšali.

Če bolj natanko pogledamo funkciji `izrisi_osnovnico` in `izrisi_vmesnico`, vidimo da se razlikujeta samo v tem kaj katera funkcija izpiše – ena „`+`“ in „`-`“, druga pa „`|`“ in „ “).

V programih ni priporočljivo imeti dve zelo podobni kodi na dveh različnih koncih. Zato bomo napisali novo funkcijo `izrisi_daljico` in jo bost auporabljali obe naši funkciji za izris osnovnice in vmesnice. Pri definiranju te funkcije si bomo pomagali z *uporabo parametrov*. Funkcija bo imela tri parametre:

- znak na krajiščih daljice,
- znak med krajiščema daljice in
- koliko znakov bo dolga daljica.

Znaka prenesemo kot spremenljivki tipa `char`.

Osnovnico bomo izrisali, tako da bomo klicali funkcijo `izrisi_daljico`. Podobno bomo izrisali tudi vmesnico. Ker smo napisali novo funkcijo, nam ni več potrebno imeti tako obsežnih in zapletenih funkcij za izris osnovnice in vmesnice. Zato ju lahko skrčimo.

Kvadrat zapolnimo z „`*`“ tako, da pri izrisu vmesnice daljico izrišemo z znakom „`*`“ namesto z „ “.

Pravokotnik

Poskusimo sedaj naš kvadrat spremeniti v poljuben pravokotnik. To izvedemo tako, da najprej prebermo širino in dolžino in nato bomo še narisali pravokotnik.

Dolžino in širino preberemo s funkcijo `preberi_sirino`. Ko program prevedemo, nas bo dvakrat vprašal po dolžini stranice, kjer bo prvič mišljena širina in drufič dolžina pravokotnika. To dvoumje moramo popraviti.

Popravimo ga lahko tako, da napišemo dve funkciji: eno, ki bo izpisala `Vpisi sirino` in drugo, ki bo izpisala `Vpisi sirino`. Ker se ti funkciji razlikujeta samo v bedsedah `sirino` in `visino`, bomo definirali le eno funkcijo (razlog glej zgoraj pri definiciji funkcije `izpisi_daljico`), ki bo kot parameter imela niz (*string*), katerega vrednost bo bodisi `visino` ali `sirino`.

Tako je v glavo funkcije `preberi_velikost` potrebno dodati parameter `char * velikost`. Sedaj moramo v isti funkciji še malce popraviti klic funkcije `printf` in sicer tako, da nas bo uporabil spremenljivko `velikost`, ki je parameter funkcije.

Kako izgleda celoten program izgleda, si lahko ogledate na našem podreočju pod imenom `pravokotnik.c`.