

PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 21 (1993/1994)

Številka 1

Strani 16-20

Jože Marinček:

MOJ PRVI GUMB

Ključne besede: računalništvo.

Elektronska verzija: <http://www.presek.si/21/1160-Marincek.pdf>

© 1993 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

RAČUNALNIŠTVO

MOJ PRVI GUMB

Programski jezik BASIC je bil pred leti zelo priljubljen, zlasti pri začetnikih. Odlikovala ga je preprosta uporaba. Sama beseda basic v angleščini pomeni osnoven, enostaven. Jezik pa je dobil ime po začetnicah Beginner's All-purpose Symbolic Instruction Set, kar bi prevedli kot simbolični jezik za splošno uporabo, namenjen začetniku. Enostavnost je imela korenine v dejstvu, da je BASIC v osnovi tolmač. Torej mu damo ukaz in on ga izpolni. Programi v BASIC-u pa so zaporedja takih ukazov. Priljubljenost BASIC-a je dosegla vrhunec s pojavom hišnih računalnikov. Kasneje je zaradi nekaterih slabosti njegova zvezda nekoliko zašla. Prva pomanjkljivost je počasnost, saj tolmač znova in znova sproti razvozlava naše ukaze. Druga šibka točka je množica dialektov, ki so nastali ob poskusih, da bi jezik izboljšali. Kot pošten Ljubljančan ne razume vsega, kar pove Prekmurec, in obratno, tako BASIC enega proizvajalca ne razume vsega, kar razume BASIC drugega proizvajalca (in obratno).

Dandanes so v naših domovih vedno bolj razširjeni tako imenovani osebni računalniki. Med njimi pa se nezadržno širi operacijski sistem MS Windows (Okna). Odlika dela z Okni je enostavnost, saj kar z miško pokažemo, kaj pričakujemo od škatle pred nami. Na hitro ponovimo, kako delamo z Windowsi. Če hočemo *pokazati* neki objekt, z miško pokažemo nanj in pritisnemo na njeno (ponavadi levo) tipko. Včasih pomaga, če pritisnemo dvakrat zapored. Najpogosteje pritiskamo na *gumbe* različnih vrst. Kadar moramo z miško kaj *povleči*, to storimo tako, da premikamo miš in hkrati držimo pritisnjeno (levo) tipko. Besedilo *vnašamo* v posebne škatlice, v katerih se miškina puščica spremeni v navpično črto. Pogosto lahko odgovor izberemo iz *seznama* pravih odgovorov.

Prav zaradi tega pa je programiranje v okolju Windows še toliko bolj zapleteno. Na prvi pogled se celo zdi, da je začetniku povsem nepristopno, ker moramo uporabljati gromozansko število funkcij, da lahko nadzorujemo okna, miško in druge stvari. Na srečo pa nam Visual Basic omogoča tudi drugi pogled. In ta je prav presenetljiv.

Prednosti Visual Basica pred ostalimi jeziki, ki so doma v Windowsih, so podobne prejšnji dobrim lastnostim BASIC-a. Zopet gre za neke vrste tolmač. Je poceni, okrnjena verzija celo zastoj. In, kar je za nas najpomembnejše, prav zabavno ga je uporabljati.

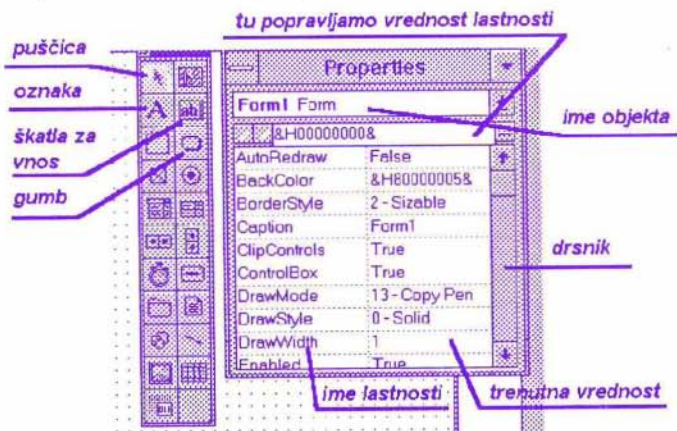
Namesto programa imamo v Visual Basicu *projekt*. To je zbirka obrazcev, kontrolnih elementov in ustreznih podprogramov.

Osnova projekta je *okvir* (Form). Nanj lahko nalepimo gumbe, škatlice za vnos besedila, sličice, sezname, besedila in sploh vse *kontrolne elemente* (Controls), ki jih uporabljajo vsi Windows programi, vključno s samimi Windowsi. V resnici je okvir *objekt*, ki ima določene lastnosti, ki mu jih lahko spremenimo. Prav tako so objekti vsi zgoraj navedeni elementi. Programiranje v Visual Basicu ni nič drugega kot spreminjanje lastnosti in metod teh osnovnih objektov.

Kontrolni elementi so zopet objekti. Z njihovo pomočjo vnašamo podatke in izpisujemo rezultate ter opravljamo še druge naloge. Primeri kontrolnih elementov so gumb (ki ga lahko pritisnemo), besedilo (izpiše besedilo), slika (nariše sličico) in drugi.

S *podprogrami* določamo, kako vsi ti objekti sodelujejo med seboj in se odzivajo na različne dogodke. Podprogrami so edino, kar še spominja na BASIC, kot ga poznamo denimo s Sinclairjevih mlinčkov.

Ko pokličemo Visual Basic, se po celem zaslonu odpre množica oken. Osrednje okno je okvir z naslovom Form 1. Drugo okno je seznam lastnosti objekta (Properties), ki ga trenutno določamo. Na začetku so v tem oknu nastete lastnosti okvira, kot so barva ozadja, oblika črk, napis na vrhu (naslov) in še vrsta drugih. Za začetek okviru spremenimo naslov. Z miško pokažemo na polje, kjer piše Caption. Ta lastnost določi ime okvira. (Če lastnosti ne vidimo, jo poiščemo s pomočjo drsnikov.) Spremenimo jo v polju na vrhu seznama lastnosti (glej sliko 1). Vtipkamo npr. "Moj prvi program". Naslov okvirja se spreminja hkrati s tipkanjem.

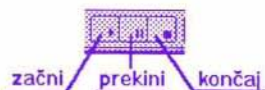


Slika 1. Škatla z orodji in seznam lastnosti

Naslednje okno je škatlica z orodji (Toolbox). Tam imamo zbrane vse kontrolne elemente, ki jih pozna Visual Basic. Delo z njimi je zelo enostavno. Najprej pokažemo objekt, s katerim želimo delati. Nato se pomaknemo na okvir in tam narišemo objekt. Potem določimo njegove lastnosti. Pri tem puščica pa pomeni, da ne bomo dodajali novega objekta, ampak bomo pokazali objekt, ki je že na obrazcu, in spremenili nekatere njegove lastnosti.

Za osnovni primer bomo na našem okvirju dodali besedilo. Z miško pokažemo polje, kjer je narisana črka A. Potem v okvirju pomaknemo miško tja, kjer hočemo imeti vogal besedila, in z njo potegnemo do nasprotnega vogala. V oknu z lastnostmi se sedaj pokažejo lastnosti besedila. Poiščemo lastnost `Caption` in jo spremenimo (podobno kot smo spremenili naslov okvirja), na primer v "Dober dan!" (naš program mora biti kar se da prijazen). Na enak način lahko na okvir prillepimo poljubni kontrolni element.

Sestavili smo program, ki izpiše "Dober dan!". Kako to preverimo? V četrtem oknu, kjer so zbrani ukazi Visual Basica (in je običajno na vrhu zaslona), poiščemo gumb `START`, ki požene program. Res, to je gumb, ki spominja na gumb za predvajanje kasete v vašem radiu. Ko ga pritisnemo, pomožna okna (škatlica z orodji, seznam lastnosti) izginejo, prav tako pa tudi pikice na okvirju, ki so namenjene lažji postavitvi objektov nanj. Na okvirju se izpiše naše besedilo, program pa čaka, da ga ustavimo. Seveda bomo v ta namen uporabili gumb `STOP`, ki je spet tak kot na radiu.



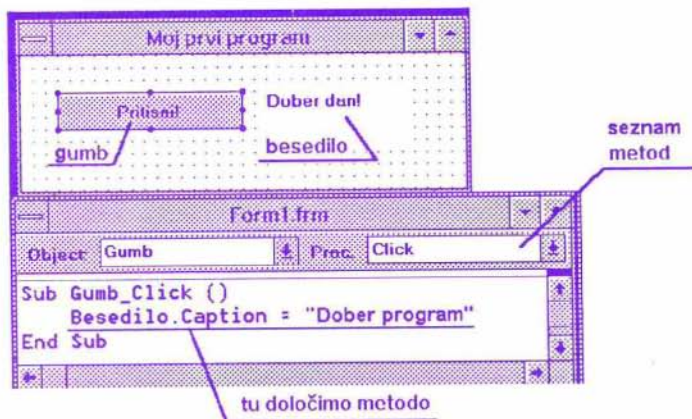
Slika 2. Gumba `START`, `STOP`

Vendar program, ki samo izpiše besedilo, ni pretirano zanimiv. Pa ga dopolnimo. Ko smo ustavili program, se je zopet pokazala škatlica z orodji. Izberemo puščico in pokažemo besedilo. Okoli besedila se pojavijo majhni kvadratici, kar pomeni, da ta objekt sedaj spreminjamo. V oknu z lastnostmi poiščemo lastnost `Name` (ime). Trenutno je besedilu ime "Label 1". Ime popravimo na "Besedilo". Potem med orodji izberemo gumb in ga dodamo na okvir. Popravimo mu ime (lastnost `Name`) na "Gumb" in napis (lastnost `Caption`) na "Pritisni!". (Razlika med imenom in napisom je, da je napis namenjen nam, ime pa drugim kontrolnim elementom, kot bomo spoznali čez

nekaj vrstic.) Sedaj bi radi dosegli še, da se bo ob pritisku na gumb napis v besedilu iz "Dober dan!" spremenil v "Dober program.". To lahko naredimo s pomočjo *metod*. Metode so aktivne lastnosti gumba. Ko se gumbu nekaj zgodi (ga pritisnemo, dvakrat pritisnemo, ipd.), se sproži ustrezna metoda in pokliče podprogram. Ta pa naredi, kar se nam zdi primerno. Metode poznajo vsi kontrolni elementi, ne samo gumbi. Do seznama metod, ki jih pozna določen objekt, pridemo tako, da pokažemo nanj in dvakrat pritisnemo z miško. Odpre se še eno okno. Šele v to okno lahko pišemo podprogram v Visual Basicu, ki predstavlja določeno metodo. V našem primeru torej pokažemo gumb in dvakrat pritisnemo. Iz seznama metod izberemo pritisk (Click). Ker še nismo povedali, kaj naj se zgodi ob pritisku na gumb, je ta metoda prazna (točneje, vsebuje le stavka za začetek in konec podprograma). V prazno vrstico med začetek in konec vnesemo:

```
Besedilo.Caption = "Dober program."
```

(glej tudi sliko 3).



Slika 3. Metoda Click

"Besedilo" je ime našega besedila, Caption pa je lastnost, ki določa, kaj to besedilo izpiše. Na ta način lahko poznamo vsako lastnost objektov, le vedeti moramo za njihovo ime. Sedaj lahko nemudoma zopet pritisnemo tipko START. Besedilo izpiše "Dober dan.". Pritisnemo gumb in ...

Tako smo brez težav napisali prvi program v Visual Basicu. Prav zaradi te enostavnosti pa Visual Basic vedno bolj uporabljajo tudi za resno programi-

