

Vizualno programiranje

Dražen Lučanin, 0036428867
mentor: doc. dr. sc. Domagoj Jakobović
smjer: računarska znanost

2009./2010. ljetni semestar

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

Osobine vizualnih programskih jezika

Problemi prikaza

Problemi programskog jezika

Zaključak

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

Osobine vizualnih programskih jezika

Problemi prikaza

Problemi programskog jezika

Zaključak

Vizualno programiranje - alternativa klasičnom programiranju orijentiranom na sintaksu i pisanje teksta

Skraćenice:

- ▶ VP - vizualno programiranje
- ▶ VPJ - vizualni programski jezik
- ▶ KPJ - klasični programski jezik
- ▶ OVP - okolina za vizualno programiranje

Pisanje teksta - od ...

"To be, or not to be: that is the question: Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to take arms against a sea of troubles, And by opposing end them? To die: to sleep; No more; and by a sleep to say we end..." -
William Shakespeare

Pisanje teksta - ... pa sve do

```
#include <stdio.h>
int main(){
    int i=0;
    while (1){
        if (i%2==0)
            printf("To be");
        else
            printf(" or not to be\n");

        ++i;
    }
    return 0;
}
```

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

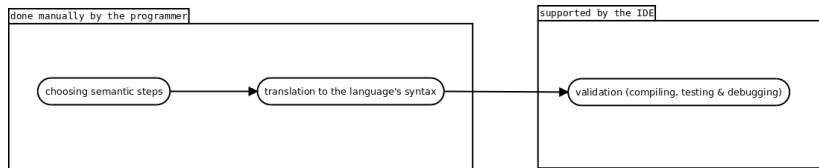
Osobine vizualnih programskih jezika

Problemi prikaza

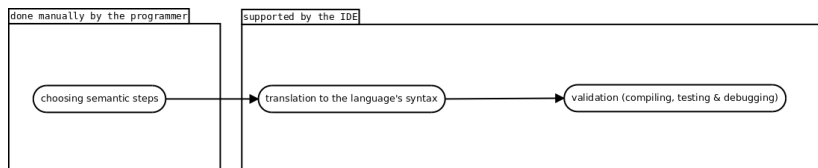
Problemi programskog jezika

Zaključak

Metoda klasičnog programiranja



Metoda vizualnog programiranja



Definicija

VPJ - svaki programski jezik u kojem korisnik stvara programe upravljajući grafičkim elementima umjesto da ih zapisuje u tekstu

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

Osobine vizualnih programskih jezika

Problemi prikaza

Problemi programskog jezika

Zaključak

Labview - istraživačka i komercijalna primjena

The image displays two windows from the LabVIEW software interface. The top window is the front panel, titled "Generate, Process, Analyze and Display Signals". It features four rotary controls for "User Operation Inputs": Signal Frequency (range 0-20), Signal Amplitude (range -6 to 6), Noise Amplitude (range 0-5), and Trace Separation (range 0-8). A "STOP EXECUTION" button is located at the bottom. On the right, there are two plots: "Time Domain Display" and "Frequency Domain Display". Each plot shows two traces: "Signal & Noise" (a noisy yellow waveform) and "Pure Signal" (a clean red sine wave). The Time Domain plot has Amplitude on the y-axis (-10 to 10) and Time on the x-axis (0 to 0.5). The Frequency Domain plot has Amplitude on the y-axis (0.0001 to 10) and Frequency on the x-axis (0 to 50).

The bottom window is the block diagram, titled "Block Diagram (LabVIEW 'G'-Source Code)". It shows a "Main While Loop: keeps the VI running until the Stop button is pressed". Inside the loop, four subVIs are used: "Trace Separation", "Signal Frequency", "Signal Amplitude", and "Noise Amplitude". These are followed by "Generate Sine Wave" and "Generate Noise" subVIs. The outputs are combined and sent to "Time Domain Display" and "Frequency Domain Display" subVIs. A "Calculate FFT of the pure sine wave and the one with noise" subVI is also present. A "stop" button is connected to the loop's exit condition.

This VI continuously generates two signals: a pure sine wave of variable frequency and amplitude and a white noise signal of variable amplitude. The noise is then added to the pure sine. The sine wave with and without the noise are then shown in a time domain graph. Additionally an FFT is calculated for both signals and the results are then shown in the frequency domain graph. Note that the square shaped functions are subroutines in the form of subVIs.

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

Osobine vizualnih programskih jezika

Problemi prikaza

Problemi programskog jezika

Zaključak

Korisnost VPJ

- ▶ želimo moći ostvariti što više toga: mali \leftrightarrow veliki projekti
- ▶ problem prilagodljivosti (engl. *scaling-up problem*)
 - ▶ KPJ - OK
 - ▶ VPJ - ?

Statički prikaz

- ▶ program "u mirovanju"
- ▶ tipično
 - ▶ dijagram toka
 - ▶ prijelazi stanja
- ▶ problematični 4D VPJ - npr. *programming-by-demonstration*
- ▶ poželjno
 - ▶ lako uređivanje
 - ▶ skrivanje suvišnih detalja

Ekran, dokumentacija

- ▶ želimo učinkovito koristiti ekran
- ▶ prednost - VPO ima dobar uvid semantiku
- ▶ dokumentacija - mogućnost za nadmašivanje KPJ
 - ▶ drag'n'drop elemenata
 - ▶ mouse-over prikaz oblačića
 - ▶ brzo reagiranje VPO - računanje izlaza u realnom vremenu

Problemi programskog jezika

- ▶ apstrakcija procedurama - "otvaranje" elemenata
- ▶ apstrakcija podataka - slikovni prikazi
- ▶ tipovi podatka - kao KPJ, poželjno dinamičko prepoznavanje
- ▶ perzistencija podataka - može preuzeti VPO

Sadržaj

Uvod

Ideja vizualnog programiranja

Primjeri

Osobine vizualnih programskih jezika

- Problemi prikaza

- Problemi programskog jezika

Zaključak

Vizualni programski jezici:

- ▶ prikladni za početnike
- ▶ primjenjivi u praksi
- ▶ pogled s više razine apstrakcije
- ▶ moć VPJ \sim moć KPJ