

Срђан Дамјановић*

Универзитет у Источном Сарајеву
Факултет пословне економије Бијељина

УДК 004.432.2 VEE PRO
DOI 10.7251 / NSK1501001D
Оригинални научни рад

СИМУЛАЦИЈА ЛИНИЈСКИХ И ЦИКЛИЧНИХ ПРОГРАМСКИХ СТРУКТУРА У ПРОГРАМСКОМ ЈЕЗИКУ VEE PRO

***Апстракт:** У овом раду су представљени примјери симулације линијских и цикличних програмских структура на рачунару из предмета Принципи програмирања. Симулација је урађена у програмском језику VEE Pro. VEE Pro је линијски програмски језик првенствено намијењен за програмско повезивање рачунара са Hewlett Packard (односно данас Agilent) мјерним инструментима, али и програмабилним мјерним инструментима осталих најпознатијих свјетских произвођача. У раду су представљене основе програмског језика VEE Pro. Због своје једноставности у писању програма и лакоћу праћењу тока извршења програма, овај програмски језик може да послужи почетницима за савладавање првих корака у програмирању.*

У овом раду је представљена симулација неколико основних програмских структура, које се изучавају на првим часовима упознавања са основама програмирања. Представљене су симулације просте линијске, разгранате и цикличне програмске структуре. Поменуте симулације се могу уз мале дораде користити у настави и у основним и средњим школама на часовима Информатике. Програмски језик VEE Pro би се могао искористити и за симулацију лабораторијских вјежби из предмета Основи електротехнике на Педагошком факултету у Бијељини.

***Кључне ријечи:** програмски језик, симулација, структура, учење, школа.*

Увод

VEE Pro је линијски програмски језик првенствено намијењен за програмско повезивање рачунара са Hewlett Packard (односно данас Agilent) мјерним инструментима, али и програмабилним мјерним инструментима осталих најпознатијих свјетских произвођача. Због своје једноставности у

* srdamjan@yahoo.com

писању програма и лако праћењу тока извршења програма, овај програмски језик може да послужи почетницима у програмирању за савладавање првих корака у програмирању. Овај програмски језик до сада има више успјешних верзија програма. Основне предности употребе програмског језика VEE Pro су:

- 1) Једноставност до крајњих граница што је једна од особина која га издваја од осталих развојних програмских пакета;
- 2) Велики број готових и разноликих објеката за управљање;
- 3) Функционалан и разнолик графички интерфејс за унос и приказ података;
- 4) Виртуелни генератори функција, импулса и шума са промјенљивом амплитудом и фреквенцијом;
- 5) Подржава комуникацију са свим програмима из Windows окружења, јер су све његове функције и сервиси на располагању, тако да се не губи вријеме на програмирање онога што Windows већ садржи;
- 6) Могућност повезивања са свим Hewlett Packard (односно данас Agilent) мјерних инструментима, али и програмабилним мјерним инструментима осталих најпознатијих свјетских произвођача.

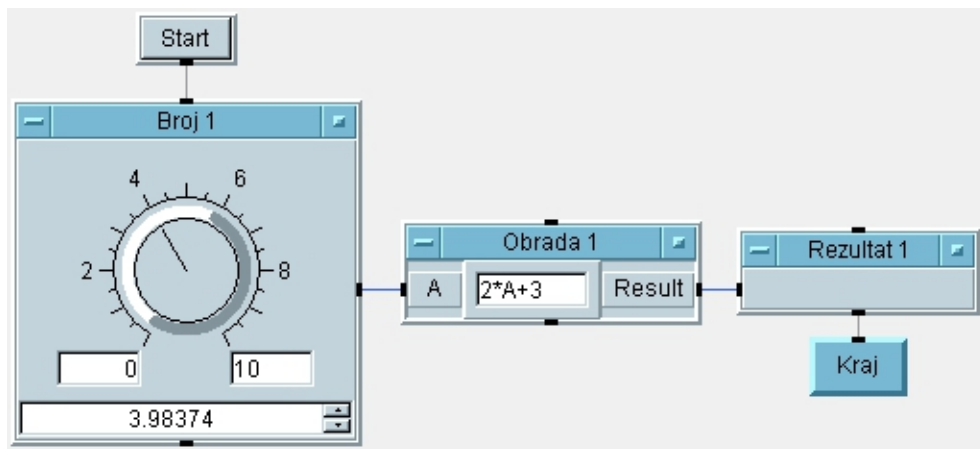
Програмски језик VEE Pro може да послужи за писање програма, који представљају симулацију на рачунару лабораторијских вјежби из предмета Основи електротехнике. Ово је посебно интересантно за студенте на Педагошком факултету у Бијељини, који на располагању немају лабораторију са мјерним инструментима. Генератори функција, генератори импулса, генератори шума, калибратори, осцилоскопи, спектрални анализатори и дигитални мултиметри су изузетно скупи мјерни инструменти, да би се у лабораторији на факултету поставило 10 радних мјеста са овим инструментима. У програмском језику VEE Pro постоје објекти, који вјерно симулирају ове инструменте, чија је јединична цијена од неколико стотина па до неколико десетина хиљада конвертибилних марака.

Методологија

Главни циљ овог рада је да се представе могућности коришћења програмског језика VEE Pro приликом извођења теријске наставе из предмета Принципи програмирања на Педагошком факултету у Бијељини. Један дио студената који слушају наставу из предмета Принципи програмирања се тек на факултету први пут срећу са прављењем алгорита и програмирањем. Студентима је поготово тешко да схвате и разумију разлику између линијских и цикличних програмских структура, као и појам гранања у програму. Због тих студената је у програмском језику VEE Pro припремљено

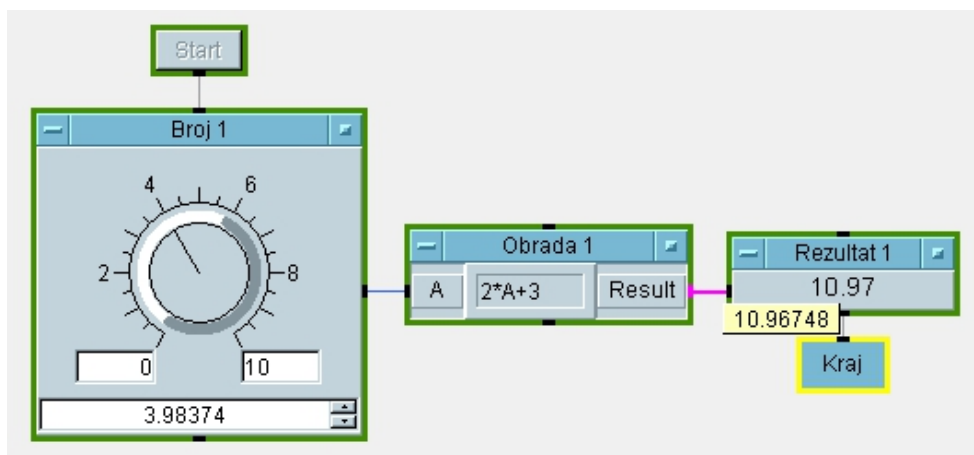
неколико симулација поменутих програмских структура. Приликом писања програма у програмском језику VEE Pro користе се готови програмски блокови у виду правоугаоника. Ови програмски блокови се повезују линијама, које уједно одређују и ток извршавања програма. На крају програм умногоме подсјећа на алгоритам, представљен преко стандардног дијаграма тока. Главна мана алгоритма је што се он не може покренути на рачунару, тако да студенти не могу да виде и провјере резултат који су добили. Симулацијом алгоритма преко програмског језика VEE Pro студенти могу да виде резултате који се очекују од алгоритма. Друга велика предност коришћења ових симулација је што се програм може извршавати корак по корак, од једног до другог програмског блока. При томе нема временског ограничења за задржавање на појединим блоковима. На тај начин се може радити детаљна анализа резултата обраде сваког програмског блока.

На Слици 1. приказана је симулација просте линијске структуре. Карактеристично за просте линијске структуре је да се сваки блок ове структуре извршава само једном, при једном покретању програма. Први елемент програма је блок **Start**, који представља почетак програма. Други елемент је **Blok 1**, који представља симулацију потенциометра, преко кога се генеришу бројеви у распону од 0 до 10. Трећи блок **Obrada 1** представља блок обраде у коме се улазна вриједност у овај блок множи са 2 и на ту вриједност додаје се број 3. Четврти блок **Rezultat 1** служи за приказ резултата обраде. Задњи блок **Kraj** означава крај програма.



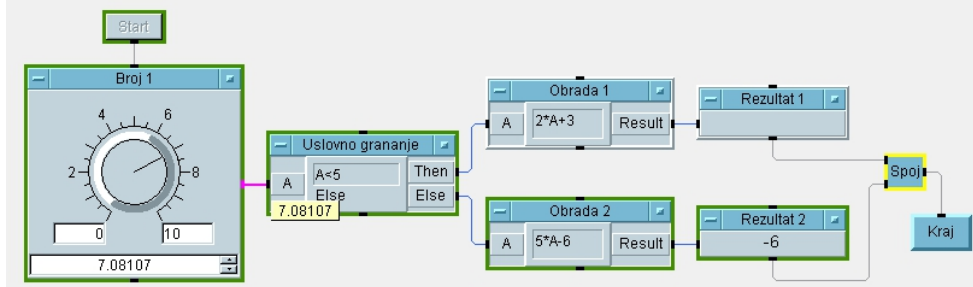
Слика 1. Изглед програма просте линијске структуре прије покретања

На Слици 2. приказано је како изгледа програм, када се извршава корак по корак. Жутим оквиром се означава блок до кога је стигло извршење програма (блок **Kraj**). Доласком миша на линију која спаја два блока у сваком тренутку се може проверити вриједност коју програм има на крају тог блока. На овој слици се види да је иза блока **Obrada 1** резултат обраде 10.96748, а да се у слиједећем блоку за приказ резултата тај број заокружује на два децимална мјеста.

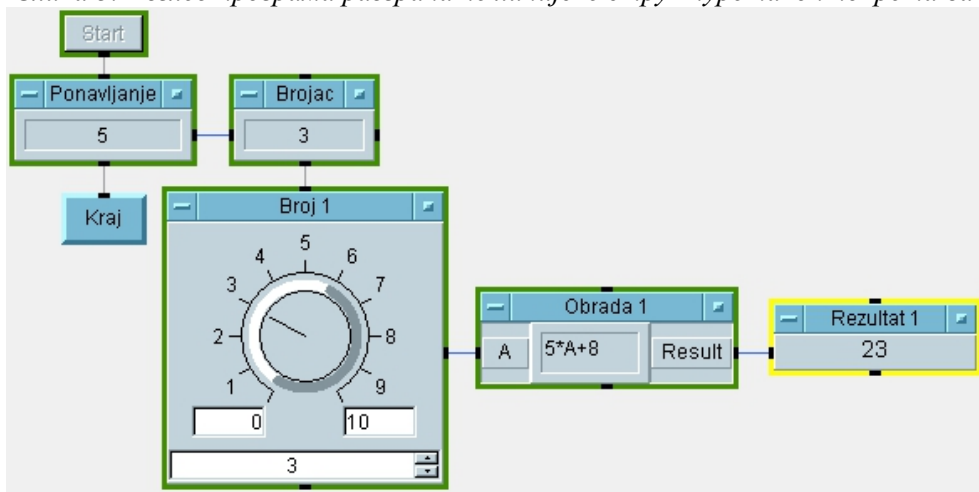


Слика 2. Изглед програма просте линијске структуре након покретања

На Слици 3. приказана је симулација једне разгранате линијске структуре. Разграната линијска структура је карактеристична по једном или више блокова условног гранања. Ако је услов гранања у блоку **Uslovno grananje** испуњен, онда се даљи ток програма одвија кроз два горња блока **Obrada 1** и **Rezultat 1**, који чине једну просту линијску структуру. Ако услов гранања на слици није испуњен, онда се даљи ток програма одвија кроз два доња блока **Obrada 2** и **Rezultat 2**, који чине другу просту линијску структуру. У примјеру са Сlike 3. није испуњен услов гранања у блоку **Uslovno grananje**, па се програм извршава преко доње линијске структуре, док се горња линијска структура неће уопште извршити.



Слика 3. Изглед програма разгранате линијске структуре након покретања



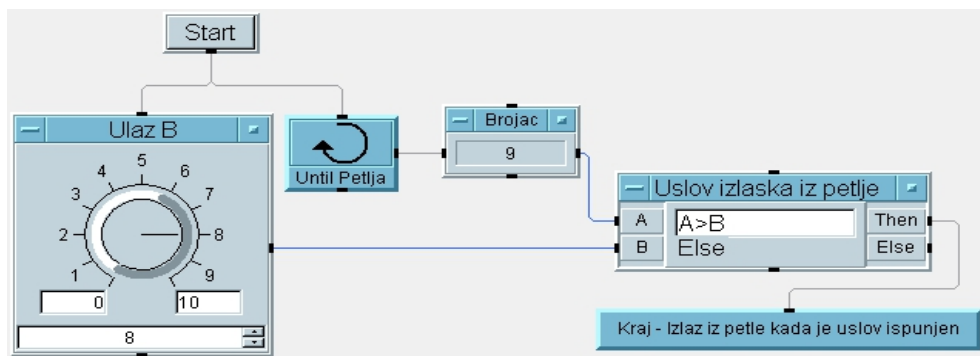
Слика 4. Изглед програма цикличне структуре након 3 циклуса

Низ алгоритамских корака у којем се један или више алгоритамских корака може извршити више од једанпут, при једном извршавању алгоритма, представља цикличну структуру или циклус. На Слици 4. приказана је циклична програмска структура, код које је блоку **Ponavljjanje**, постављено да се пет пута понавља линеарна циклична структура **Brojac**, **Broj 1**, **Obrada 1** и **Rezultat1**. Крај програма наступа када се поменути блок изврши пет пута, што је приказано на Слици 5.

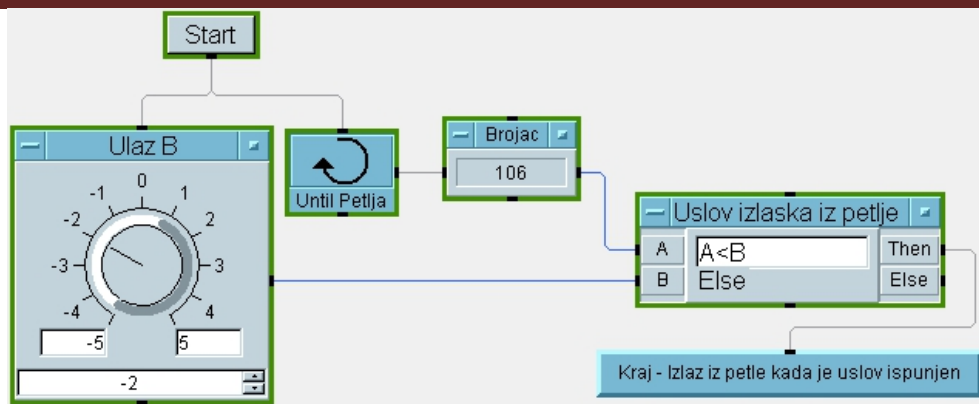


Слика 5. Изглед програма цикличне структуре након 5 циклуса

На Сlici 6. приказана је симулација UNTIL петље. Ова програмска структура је карактеристична по томе што се неки програмски блокови понављају све док не буде испуњен логички услов за излазак из петље, постављен у блоку **Uslov izlaska iz petlje**. UNTIL петља је карактеристична по томе што се може десити да логички услов никада не буде испуњен. У том случају би се петља бесконачно понављала и програм се не би никада завршио. То представља велику опасност за програмере, јер доводи до блокирања рада рачунара. На Сlici 7. је приказана UNTIL петља, која се никада неће завршити, јер никада неће бити задовољен логички услов за излазак из петље.



Слика 6. Изглед програма са петљом која има излаз на крају



Слика 7. Изглед програма са петљом која нема краја

На сличан начин се може направити програм за симулацију петље са излазом на врху петље (While петља).

Закључак

Основни допринос који се очекује од примјене симулације линијских и цикличних програмских структура на рачунару из предмета Принципи програмирања на Педагошком факултету у Бијељини је упознавање студената са логиком функционисања програма. Ово је посебно значајно за студенте који се први пут срећу са програмирањем у свом школовању. За професоре је изазов да управо те студенте заинтересује за овај предмет. Управо су ове симулације оно што треба да разбије сувопарност предавања, када студенти виде алгоритамске шеме на табли. Студентима су алгоритми често нејасни јер не могу да их покрену на рачунару да би добили очекивани резултат. Кроз ове припремљене симулације студенти се настоје провести кроз програмске структуре од почетка до краја кроз све програмске блокове. При томе се визуелно врло лако уочава који се програмски блок тренутно извршава, који блок му је претходио и који је наредни програмски блок који ће се извршити. Заустављањем програма на сваком програмском блоку може се посебно детаљно појаснити разграната и циклична програмска структура. Лако се могу мијењати логички услови у блоковима гранања и практично визуелно показати у ком смјеру ће се програм даље извршавати.

Програмски језик VEE Pro се показао као врло једноставан за учење, а са друге стране посједује велики број готових објеката, који представљају симулацију класичних програмских блокова. Међутим, овај програмски језик има и велики број готових блокова, који симулирају разне мјерне

инструменте. То су објекти који симулирају аналогне и дигиталне мултиметре, генераторе функција, генераторе импулса, генераторе шума, калибраторе, осцилоскопе, спектралне анализаторе и друге. Ови инструменти су у пракси углавном вишефункцијски и имају високу цијену, тако да факултети углавном данас нису у могућности да набаве више ових мјерних инструмената за потребе извођења лабораторијских вјежби. Руковање са овим реалним инструментима је често компликовано, јер се од руковоаца тражи детаљно упознавање са техничком документацијом, а за то је потребно доста времена као и искуства. Студенти Педагошког факултета на смјеру наставника Техничког образовања и Информатике обично немају довољно знања да би могли успјешно да савладају рад са овим инструментима. Са друге стране, рад у програму са објектима који представљају симулацију ових реалних инструмената је прилично једноставан. Чак и у случају када студент није довољно спреман за неку вјежбу и у раду направи грешку, та грешка ће бити само у програму и то неће проузроковати велику материјалну штету.

Друга предност ових лабораторијских вјежби је што се оне могу изводити на било ком рачунару, у било које вријеме и што се могу понављати велики број пута, како би студент што боље савладао неке проблеме. На овај начин факултет може да уштеди значајна финансијска средства, а да се квалитет наставе и знање студената подигну на значајно виши ниво.

Надамо се да ће студенти бити заинтересовани да кроз пројекте и дипломске радове уз помоћ програмског језика VEE Pro припремају нове симулације програмских структура.

Литература

- Дамјановић, С., Катанић, П. (2011). *Програмски језик VEE PRO*. Источно Сарајево: Електротехнички факултет.
- Дамјановић, С., Катанић, П. (2014). *Програмски језик VEE PRO*. Бијељина: Факултет пословне економије.
- Дамјановић, С., Катанић, П., Поповић, Б. (2011). Симулација лабораторијских вјежби из предмета електрична мјерења у програмском језику *VEE Pro*. *Зборник радова са Симпозијума ИНФОТЕХ 2011*, Е-V-10, 793 – 798.

Srđan Damjanović

Simulation of Linear and Cyclic Program Structures in the Programming Language VEE PRO

Summary

This paper presents the examples of computational simulations of linear and cyclical programming structures within the subject of programming principles. Simulations were carried out in the programming language VEE Pro. The VEE Pro language is primarily utilized to enable the interfacing of computers with Hewlett Packard (now Agilent) measuring instruments, as well as programmable measuring instruments from other world manufacturers. This paper presents the basics of the programming language VEE Pro. Due to the simplicity in executing and analyzing programs, VEE Pro is well suited for programming beginners.

This paper presents several simulations of basic programming structures, which are routinely taught in an introductory course. The mentioned simulations can be modified to suit an elementary or middle school settings where Computer Science is taught. The VEE Pro programming language can also be used for the simulation of laboratory practices in Fundamentals of Electrical Engineering course at the Faculty of Education in Bijeljina.

Key words: *programming language, simulations, structures, learning, school*