

ФАКУЛТЕТ ЗА  
ПРИМЕЊЕНИ  
МЕНАџМЕНТ  
ЕКОНОМИЈУ  
И ФИНАНСИЈЕ



# MEFKON

International Scientific Conference

МЕЂУНАРОДНА НАУЧНО-СТРУЧНА КОНФЕРЕНЦИЈА

## INNOVATION AS AN INITIATOR OF THE DEVELOPMENT ИНОВАЦИЈЕ КАО ПОКРЕТАЧ РАЗВОЈА

INNOVATIVE ACTIVITIES – CONTEMPORARY CHALLENGES AND SOLUTIONS  
ИНОВАТИВНА ДЕЛАТНОСТ – САВРЕМЕНИ ИЗАЗОВИ И РЕШЕЊА

INTERNATIONAL CONFERENCE PROCEEDINGS

ЗБОРНИК РАДОВА СА МЕЂУНАРОДНОГ СКУПА

# INNOVATIONS

5. децембар 2019. Београд

Универзитет Привредна академија у Новом Саду

University Business Academy in Novi Sad

Факултет за примењени менаџмент, економију и финансије Београд

Faculty of Applied Management, Economics and Finance Belgrade

Међународна научно-стручна конференција

International Scientific & Professional Conference

**МЕФкон 2019 / MEFkon 2019**

*ИНОВАЦИЈЕ КАО ПОКРЕТАЧ РАЗВОЈА*

„Иновативна делатност – савремени изазови и решења“

*INNOVATION AS AN INITIATOR OF THE DEVELOPMENT*

“Innovative Activities – Contemporary Challenges and  
Solutions”

ЗБОРНИК РАДОВА СА МЕЂУНАРОДНОГ СКУПА

INTERNATIONAL CONFERENCE PROCEEDINGS

Београд, 5. децембар 2019. године  
Међународна научно-стручна конференција  
**МЕФкон 2019:**

„Иновације као покретач развоја“  
Зборник радова са међународног скупа –  
*електронско издање*  
*\*радови су објављени у изворном облику*

Belgrade, December 5<sup>th</sup> 2019  
International Scientific & Professional Conference  
**MEFkon 2019:**

“Innovation as an Initiator of the Development”  
International Conference Proceedings –  
*digital edition*  
*\*papers were published in the original form*

Издавач / Publisher

Факултет за примењени менаџмент, економију и финансије Београд  
Faculty of Applied Management, Economics and Finance Belgrade

За издавача / For the Publisher:

Miodrag Brzaković, PhD, Council President  
Tomislav Brzaković, PhD, Dean

Уредници / Editors

Darjan Karabašević, PhD  
Svetlana Vukotić, PhD

Технички уредници / Technical editors

Sanja Anastasija Marković, MSc  
Vuk Mirčetić, MSc

Дизајн / Design

Strahinja Vidojević, Bsc

Штампа / Print

Факултет за примењени менаџмент, економију и финансије Београд  
Faculty of Applied Management, Economics and Finance Belgrade

Тираж / Number of copies

100

ISBN 978-86-84531-45-4

**Организатор / Organizer:**

Факултет за примењени менаџмент, економију и финансије, Београд  
Faculty of Applied Management, Economics and Finance, Belgrade

**Суорганизатори / Co-organizers:**

Higher School of Finance and Management, Russian Presidential Academy of National Economy and Public Administration (RANEPA), Moscow, Russia

Faculty of Management in Tourism and Commerce Timișoara, Christian University “Dimitrie Cantemir“ Bucharest, Romania

Faculty of Economics and Tourism “Dr. Mijo Mirković”, Juraj Dobrila University of Pula, Croatia

PAR University College, Rijeka, Croatia

University “Vitez”, Bosnia and Herzegovina

Institute of Agricultural Economics, Belgrade, Serbia

Faculty of Mechanical Engineering, Innovation Center, Belgrade, Serbia

Faculty of Hotel Management and Tourism – Vrnjačka Banja, University of Kragujevac, Serbia

Faculty of Economics in Subotica, University of Novi Sad, Serbia

Independent University Banja Luka, Bosnia and Herzegovina

National Association of Healthcare Professionals of Serbia, Serbia

Regional Chamber of Commerce of Šumadija and Pomoravlje Administrative District, Serbia

## **Научни одбор / Scientific Committee**

Marijana Carić, PhD, University Business Academy in Novi Sad, Serbia

Marko Carić, PhD, Faculty of Law, University Business Academy in Novi Sad, Serbia

Mirko Kulić, PhD, University Business Academy in Novi Sad, Serbia

Dragan Soleša, PhD, Faculty of Economics and Engineering Management, University Business Academy in Novi Sad, Serbia

Victor Palamarchuk, PhD, Higher School of Finance and Management, Russian Presidential Academy of National Economy and Public Administration, (RANEPA), Moscow, Russia

Stanislav Furta, PhD, Professor, Higher School of Finance and Management, Russian Presidential Academy of National Economy and Public Administration (RANEPA), Moscow, Russia

Marina Vvedenskaya, Higher School of Finance and Management, Russian Presidential Academy of National Economy and Public Administration (RANEPA), Moscow, Russia

Cipriana Sava, PhD, Faculty of Management in Tourism and Commerce Timișoara, Christian University "Dimitrie Cantemir" Bucharest, Romania

Marius Miculescu, PhD, Faculty of Management in Tourism and Commerce Timișoara, Christian University "Dimitrie Cantemir" Bucharest, Romania

Galina Verigina Mihailovna, PhD, Faculty of Economics, Russian Presidential Academy of National Economy and Public Administration (RANEPA), Moscow, Russia

Iva Slivar, PhD, Faculty of Economics and Tourism "dr. M. Mirković", Juraj Dobrila University of Pula, Croatia

Violeta Šugar, PhD, Faculty of Economics and Tourism "dr. M. Mirković", Juraj Dobrila University of Pula, Croatia

Darijo Jerković, PhD, University "Vitez", Bosnia and Herzegovina

Erdin Hasanbegović, PhD, University "Vitez", Bosnia and Herzegovina

Gordana Nikolić, PhD, Business School PAR, Rijeka, Croatia

Bisera Karanović, PhD, Business School PAR, Rijeka, Croatia

Branko Mihailović, PhD, Institute of Agricultural Economics, Belgrade, Serbia

Zoran Simonović, PhD, Institute of Agricultural Economics, Belgrade, Serbia

Svetlana Roljević Nikolić, PhD, Institute of Agricultural Economics, Belgrade, Serbia

Maja Đurović Petrović, PhD, Innovation Center of the Faculty of Mechanical Engineering, University of Belgrade, Serbia

Snežana Kirin, PhD, Innovation Center of the Faculty of Mechanical Engineering, University of Belgrade, Serbia

Jasmina Lozanović Šajić, PhD, Innovation Center of the Faculty of Mechanical Engineering, University of Belgrade, Serbia

Drago Cvijanović, PhD, Faculty of Hotel Management and Tourism in Vrnjačka Banja, University of Kragujevac, Serbia

Vladimir Senić, PhD, Faculty of Hotel Management and Tourism in Vrnjačka Banja, University of Kragujevac, Serbia

Pere Tumbas, PhD, Faculty of Economics in Subotica, University of Novi Sad, Serbia

Aleksandar Grubor, PhD, Faculty of Economics in Subotica, University of Novi Sad, Serbia

Zoran Kalinić, PhD, Independent University of Banja Luka, Bosnia and Herzegovina

Mirjana Stojanović, PhD, Independent University of Banja Luka, Bosnia and Herzegovina

Miodrag Vučić, PhD, National Association of healthcare professionals of Serbia, Serbia

Nebojša Vacić, PhD, National Association of healthcare professionals of Serbia, Serbia

Dragiša Stanujkić, PhD, Technical Faculty in Bor, University of Belgrade, Serbia

Ieva Meidutė-Kavaliauskienė, PhD, Faculty of Business Management, Vilnius Gediminas Technical University, Vilnius, Lithuania

Bratislav Predić, PhD, Faculty of Electronic Engineering, University of Niš, Serbia

Željko Stević, PhD, Faculty of Transport and Traffic Engineering, University of East Sarajevo, Doboj, Bosnia and Herzegovina

Dragan Pamučar, PhD, Military Academy, University of Defence, Belgrade, Serbia

Natalia Vuković, PhD, Russian State Social University, Faculty of Ecology, Moscow, Russian Federation

Milan Stamatović, PhD, Faculty of Business and Law, University Union – Nikola Tesla, Serbia

Darko Vuković, PhD, Saint Petersburg School of Economics and Management, National Research University Higher School of Economics, St. Petersburg, Russian Federation

Aleksandar Đoković, PhD, Faculty of Organizational Sciences, University of Belgrade

Aleksandra Fedajev, PhD, Technical Faculty in Bor, University of Belgrade, Serbia

Velemir Ninković, PhD, Swedish University of Agricultural Sciences, SLU, Sweden

Marija Panić, PhD, Technical Faculty in Bor, University of Belgrade, Serbia

Gabrijela Popović, PhD, Faculty of Management in Zaječar, Megatrend University Belgrade, Serbia

Hugo Van Veghel, PhD, Belgian Serbian Business Association, Belgium

Desimir Knežević, PhD, University of Priština, Serbia

Jonel Subić, PhD, Institute of Agricultural Economics, Serbia

Elez Osmani, PhD, Institute for Scientific Research, Montenegro

Nikola Ćurčić, PhD, Institute of Agricultural Economics, Belgrade, Serbia

Marina Milovanović, PhD, Faculty for Entrepreneurial Business and Real Estate Management, University Union-Nikola Tesla, Serbia

Boško Vojnović, PhD, Higher Education Institution for Agriculture, Serbia

Dejan Sekulić, PhD, Faculty of Hotel Management and Tourism in Vrnjačka Banja, University of Kragujevac, Serbia

Miodrag Brzaković, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Tomislav Brzaković, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Svetlana Vukotić, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Srđan Novaković, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Tatjana Dragičević Radičević, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Darjan Karabašević, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Ivona Brajević, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Mlađan Maksimović, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Adriana Radosavac, PhD, Faculty of Applied Management, Economics and Finance Belgrade,  
University Business Academy in Novi Sad, Serbia



## **Организациони одбор / Organizing Committee**

Pavle Radanov, PhD, President of the Committee, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Sanja Anastasija Marković, MSc, Vice-president of the Committee, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Jelena Petrović, MSc, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Aleksandar Brzaković, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Pavle Brzaković, PhD, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Vuk Mirčetić, MSc, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Goran Jocić, Msc, Faculty of Applied Management, Economics and Finance Belgrade, University Business Academy in Novi Sad, Serbia

Cipriana Sava, PhD, Faculty of Management in Tourism and Commerce Timișoara, Christian University „Dimitrie Cantemir“ Bucharest, Romania

Gheorghe Pinteală, PhD, Faculty of Management in Tourism and Commerce Timișoara, Christian University “Dimitrie Cantemir“ Bucharest, Romania

Dragana Pešić, PhD, Faculty of Hotel Management and Tourism in Vrnjačka Banja, University of Kragujevac, Serbia

Vesna Milovanović, PhD, Faculty of Hotel Management and Tourism in Vrnjačka Banja, University of Kragujevac, Serbia

## ПРЕДГОВОР

У савременом друштву, појам иновација и иновирања постао је веома значајан, у тој мери, да је у већини мисија и визија савремених компанија коришћење ове речи постало обавеза. Међутим, посматрано и шире, суштина свих развојних промена, углавном, огледа се у иновативности. Иновације су свуда око нас. То што су иновације толико присутне у целокупном подручју људске активности, намеће потребу да иновативност постане уводна тачка приликом анализе комплексности нове економије, друштва и културе у настајању, укључујући и индивидуу. Овај процес даље имплицира неминовно разматрање повратне спреге иновација и развоја. Управо отуда проистиче покретачки мотив да се Факултет за примењени менаџмент, економију и финансије из Београда заједно са суорганизаторима бави ове године на Четвртој међународној научно-стручној конференцији темом „Иновације као покретач развоја“.

Традиционално организовање овог међународног научног скупа има за циљ да покаже да иновација није само део пословне стратегије предузећа, већ да покреће економску добробит и утиче на прогрес целе једне земље.

Примерено теми и циљу научног скупа установљене су две сесије: I сесија: Иновације – темељ развоја (Тематски зборник) и II сесија: Иновативна делатност – напредак и будућност (Зборник радова са међународног скупа). Избор теме скупа и свеприсутност иновација, као и понуђени већи број тематских области утицао је да су у овој публикацији радови многих угледних универзитетских професора, истакнутих истраживача, експерата и научних радника, како из Србије, тако и из иностранства.

Зборник радова са међународног скупа, као резултат конференције, публикован је на CD-у и биће доступан широј научној јавности. Радови у овој публикацији значајно доприносе утврђивању нераскидиве везе између иновација и развоја. Истовремено смо показали да подручје иновација дефинитивно више није везано само за техничко – технолошки прогрес. У складу са тим, радови могу бити корисни како научној, тако и стручној јавности и свим заинтересованим за утицај иновација на развој.

Београд,

Децембар, 2019.

Уредници

Др Дарјан Карабашевић

Др Светлана Вукотић

## FOREWORD

In contemporary society, the notions of “innovation” and “innovating” have become very significant, that being so to an extent that, in the majority of the missions and visions of contemporary companies, the use of this word has become mandatory. From a broader perspective, too, however, the essence of all developmental changes mainly reflects in innovativeness. Innovations are all around us. The fact that innovations are, to such an extent, present in the overall field of the human activity imposes the need for innovativeness to become the introductory point in carrying out the analysis of the complexity of the newly-emerging economy, society and culture, also including an individual. This process is further implicative of the unavoidable consideration of the innovation-development feedback. Thence exactly arises the driving motive for the Faculty of Applied Management, Economics and Finance in Belgrade to deal with the foregoing, together with the co-organizers, at the Fifth International Scientific-Professional Conference, entitled “Innovation as an initiator of the development”.

This international scientific conference is traditionally organized with the aim of demonstrating that innovation is not only a part of an enterprise’s business strategy, but also drives economic wellbeing and influences the progress of one whole country.

Suitably to the theme and the goal of the scientific conference, the two sessions are established: Session 1 – Innovations – development prospects (Thematic Proceedings), and Session 2 – Innovative activities – contemporary challenges and solutions (International Conference Proceedings). The choice of the conference theme and the omnipresence of innovations, as well as the offered larger number of the thematic fields, have influenced the inclusion of the papers by many distinguished university professors, eminent researchers, experts and scientific workers both from Serbia and from abroad in this publication.

As a result of the Conference, the *Conference Proceedings* are published on CD and the same will be available to a wider scientific audience. The papers in this publication significantly contribute to the establishment of an inextricable liaison between innovations and development. Simultaneously, we have demonstrated that the field of innovations is definitely no longer only related to technical-technological progress. In accordance with that, the papers may also be beneficial to both the scientific and the professional public and to all those interested in the impact of innovations on development.

Belgrade,

December, 2019

Editors

Darjan Karabašević, PhD

Svetlana Vukotić, PhD

## САДРЖАЈ / CONTENT:

---

### ПЛЕНАРНА ПРЕДАВАЊА

#### PLENARY LECTURES

---

Ece Dođantan Çađlar Karamaşa	<b>DETERMINATION OF CRITICAL SUCCESS FACTORS FOR CREATING ENTREPRENEURIAL ECOSYSTEM IN SMART CITIES VIA NEUTROSOPHIC SETS</b>	1
Dragiša Stanujkić	<b>BLOCKCHAIN AND CRYPTO: PAST, PRESENT, AND FUTURE</b>	11

---

### РАДОВИ СА КОНФЕРЕНЦИЈЕ

#### CONFERENCE PAPERS

---

Tatjana Dragičević Radičević Milica Nestorović Mirjana Stojanović Trivanović	<b>KONCEPT CIRKULARNE EKONOMIJE</b>	21
Гордана Петровић Млађан Максимовић Дарјан Карабашевић	<b>КОНКУРЕТНОСТ СРПСКЕ ПРИВРЕДЕ НА ГЛОБАЛНОМ ТРЖИШТУ</b>	28
Наталья Б. Сафронова Михаил В. Рыбин	<b>ПРАКТИКА ВНЕДРЕНИЯ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ В ОТРАСЛЯХ ЭКОНОМИКИ РОССИЙСКОЙ ФЕДЕРАЦИИ</b>	37
Марина Викторовна Введенская Терентьева Ирина Дмитриевна Гришанин Никита Владимирович	<b>ВЛИЯНИЕ ИМИДЖА РУКОВОДИТЕЛЯ НА РЕПУТАЦИЮ БАНКА НА ПРИМЕРЕ ОЛЕГА ТИНЬКОВА</b>	41

---

Miloš Grujić Jelena Trivić	<b>IMPACT OF PENSION FUNDS ON FINANCIAL MARKETS DEVELOPMENT</b>	48
Наталья Б. Сафронова Алан Л. Абаев	<b>ПРОДВИЖЕНИЕ МОЛОДЁЖНЫХ СОЦИАЛЬНЫХ ПРОЕКТОВ ОБЩЕСТВЕННЫМИ ОРГАНИЗАЦИЯМИ</b>	57
Dušan Rajčević Aleksandar Šijan Ivona Brajević	<b>MODIFIED ACCELERATED PARTICLE SWARM OPTIMIZATION ALGORITHM FOR CONSTRAINED OPTIMIZATION</b>	61
Kristina Jauković Jocić Goran Jocić Ivona Brajević	<b>BOUNDARY CONSTRAINT HANDLING SCHEMES IN FIREFLY ALGORITHM</b>	68
Душан Рајчевић Горан Јоцић Драган Солеша	<b>ТАЈНОСТ КОМУНИКАЦИЈЕ</b>	75
Сања Максимовић Мојићевић Срђан Маричић Миодраг Брзаковић	<b>ПАМЕТНИ ГРАДОВИ И ЈоТ</b>	83
Душан Рајчевић Милена Мосић Драган Солеша	<b>АНАЛИЗА РАСПРОСТРАЊЕНИХ ПАРАДИГМИ У ПРОГРАМИРАЊУ</b>	90
Marko Filijović Pavle Radanov Verica Jovanović	<b>RAZVOJ PAMETNIH GRADOVA – BEZBEDNOSNI ASPEKT</b>	97
Saša Simić Srboljub Nikolić	<b>MULTI-CRITERIA DECISION- MAKING MODEL FOR PURCHASING MOBILE PHONE FOR OFFICIAL PURPOSES</b>	104
Aleksandra Fedajev Gabrijela Popović Dragiša Stanujkić	<b>MCDM FRAMEWORK FOR EVALUATION OF THE TOURISM DESTINATION COMPETITIVENESS</b>	112

Natalia Safronova Yulia Mikhaylova	<b>ADAPTATION OF FOREIGN STUDENTS AT RANERA AS ONE OF THE ASPECTS OF INTERNATIONAL COOPERATION</b>	120
Адриана Радосавац Немања Стојковић Жељко Ондрик	<b>ЕКОЛОШКИ (ЗЕЛЕНИ) МАРКЕТИНГ КАО ФАКТОР ЗАШТИТЕ ЖИВОТНЕ СРЕДИНЕ</b>	125
Stefan Ditrih Olgica Milošević Svetlana Marković	<b>DRUŠTVENO ODGOVORNO POSLOVANJE KAО PUT KA ODRŽIVOM RAZVOJU</b>	131
Adnan Salkić	<b>SREDSTVA ZA MIRNO RJEŠAVANJE SPOROVA U MEĐUNARODNOM PRAVU</b>	137
Вук Мирчетић Марија Јаношик Адам Малешевић	<b>ДЕТЕРМИНИСАЊЕ ЛИДЕРСТВА И КОМПАРАЦИЈА ТЕОРИЈСКИХ ПРИСТУПА</b>	146
Ана Чанак Ана Ненадић	<b>МОТИВАЦИОНИ ПОДСТИЦАЈИ ЗА УПРАВЉАЊЕ ПЕРФОРМАНСАМА САВРЕМЕНИХ ОРГАНИЗАЦИЈА</b>	156
Ана Ненадић Ана Чанак	<b>УТИЦАЈ ТИМСКОГ РАДА НА ЕФИКАСНОСТ ПОСЛОВАЊА САВРЕМЕНИХ ПРЕДУЗЕЋА</b>	165
Semina Škandro Erdin Hasanbegović	<b>PRIMJENA KREATIVNOG RAČUNOVODSTVA KOD PRAVNIH SUBJEKATA</b>	174
Nikola Radić Vlado Radić	<b>GLOBALNA INTEGRACIЈА KINESKIH TEHNOLOŠKIH LANACA VREDNOSTI</b>	182
Сузана Стојановић Јелена Станковић	<b>HEMATЕРИЈАЛНА УЛАГАЊА КАО ИЗВОР КОНКУРЕНТСКЕ ПРЕДНОСТИ</b>	192
Mirjana Stojanović Trivanović Tatjana Dragičević	<b>BANKOOSIGURANJE KAО INOVACIЈА ILI ŠANSA ZA POVEĆANJE PROFITA</b>	203

Radičević Milica Nestorović		
Слободан Васић Јасмина Секеруш	<b>УТИЦАЈ ИНОВАЦИЈА НА РАЗВОЈ КУЛТУРНОГ ТУРИЗМА</b>	209
Cornelia Petroman Diana Marin Ioan Petroman	<b>NEW OPPORTUNITIES FOR PRACTICING MOSAIC TOURISM IN TIMISOARA</b>	219
Cornelia Petroman Loredana Văduva Ioan Petroman	<b>PROPOSALS OF NEW SPECIFIC ACTIVITIES FOR TIMIS RURAL TOURISM</b>	225
Милена Подовац	<b>ПАМЕТНИ ГРАДОВИ У ФУНКЦИЈИ РАЗВОЈА ТУРИЗМА</b>	235
Јелена Стојковић Јелена Вукчевић	<b>ПОТЕНЦИЈАЛИ ЗА РАЗВОЈ МРАЧНОГ ТУРИЗМА У РЕПУБЛИЦИ СРБИЈИ НА ПРИМЕРУ СЕЛА МЕДВЕЂА, ОПШТИНА ТРСТЕНИК</b>	242
Ibrahim Obhodaš Mahir Zajmović Ivana Topić	<b>MODELIRANJE POSLOVNIH KARAKTERISTIKA PRIMJENOM INFORMACIONIH SISTEMA</b>	252
Oliver Momčilović Suzana Doljanica Dragan Doljanica	<b>INFLUENCE OF ICT AND EDUCATION ON AN ENTERPRENEUR AND HIS CAREER</b>	261
Jozo Piljić	<b>OBRAZOVANJE ZA PODUZETNIŠTVO – POKRETAČ KONKURENTNOSTI</b>	270
Mirsad Nalić	<b>INOVATIVNOST KAO FAKTOR KONKURENTNOSTI NA GLOBALNOM NIVOU I NA NIVOU BOSNE I HERCEGOVINE</b>	278
Marina Jovičević Simić Slobodan Živkucin Predrag Jovičević	<b>PRIMENA INOVACIJA KAO FAKTOR POVEĆANJA MEĐUNARODNE KONKURENTNOSTI KOMPANIJA</b>	290

Nebojša Pavlović	<b>RECOGNIZING THE FUTURE THAT HAS ALREADY STARTED</b>	296
Gheorghe Pinteală	<b>ASPECTS OF EUROPEAN COMPARATIVE MANAGEMENT</b>	302
Tanja Gavrić	<b>UPRAVLJANJE SISTEMOM NAGRAĐIVANJA U OBITELJSKIM PODUZEĆIMA</b>	307
Vojkan Bižić	<b>INTEGRISANE MARKETINŠKE KOMUNIKACIJE I SPORTSKE AKTIVNOSTI DECE SA POSEBNIM POTREBAMA U BEOGRADU</b>	318
Milan Nedeljković Jasmina Petrović Ana Nedeljković	<b>PROŠLOST, SADAŠNJOST I BUDUĆNOST SAVREMENIH MATERIJALA – SUPERLEGURA</b>	324
Milan Nedeljković Jasmina Petrović Ana Nedeljković	<b>SPECIJALNI METALNI MATERIJALI – NOVE LEGURE SA VISOKOVREDNIM OSOBINAMA</b>	334



**ПЛЕНАРНА ПРЕДАВАЊА**

**PLENARY LECTURES**

# АНАЛИЗА РАСПРОСТРАЊЕНИХ ПАРАДИГМИ У ПРОГРАМИРАЊУ

*Душан Рајчевић<sup>1</sup>, Милена Мосић<sup>2</sup>, Драган Солеша<sup>3</sup>*

<sup>1</sup>Факултет за примењени менаџмент, економију и финансије, Јеврејска 24, Београд, Србија, *dusan@mef.edu.rs*

<sup>2</sup>Факултет за примењени менаџмент, економију и финансије, Јеврејска 24, Београд, Србија, *milena@mef.edu.rs*

<sup>3</sup>Факултет за економију и инжењерски менаџмент, Цвећарска 2, Нови Сад, Србија, *dragan.solesa@fimek.edu.rs*

**Апстракт:** Постоји велики број програмских језика, начина и алата за развој програма. Неки језици су прилагођени само одређеним платформама, неки се могу користити за више њих. Овај рад има за циљ да анализира различите технике писања кода и да представи пресек кључних карактеристика сваког од њих.

**Кључне речи:** парадигма, програмирање, функције, објекти, класе, процедуре

## 1. УВОД

Рачунарски програм је скуп наредби који говори рачунару на који начин да обави одређени задатак. Текст програма који је читљив програмеру и кориснику зове се изворни код програма, док је извршна верзија програма бинарна, и може је читати и изводити само рачунар. Програмски језик је скуп лексичких, синтаксних и семантичких правила која уобличују строгу структуру писања изворног кода. Сваки програм састоји се од наредби. Наредбе се састоје, у зависности о ком програмском језику се ради, од израза, константи, идентификатора и позива. Уопштено говорећи, постоји више скупова програмских језика, и подељени су у 4 генерације језика.

Постоји велики број програмских језика, начина и алата за развој програма. Неки језици су прилагођени само одређеним платформама, неки се могу користити за више њих. Развијено је и неколико програмских техника (парадигми) за писање програма. Програмска парадигма представља поглед који програмери имају над програмом и његовим извршавањем. Парадигмом се дефинише начин писања, организације и извршавања кода како би исти био погодан како за човека који га пише, чита и разуме, тако и за рачунар који га извршава. Основна подела рачунарских парадигми је на императивну и декларативну групу. Императивна у коју спадају процедурално програмирање и објектно-оријентисано програмирање, акценат ставља на начин на који ће се задатак обавити, док декларативна (функционално и логичко) на сам задатак, шта све треба да се обави, али не и како.

## 2. ПАРАДИГМЕ

### 2.1. Процедурално програмирање

Процедурално програмирањем можда је најчешће прва парадигма са којом се програмери први пут сусретну. У основи, процедурални код директно даје упутства уређају на који начин да изврши задатак, у линеарним корацима. Ова парадигма користи линеарни приступ одозго према доле и обрађује податке и поступке као две одвојене целине. Процедурално програмирање дели програм на процедуре као функције, и у основи садржи низ корака које је потребно спровести. Оно подразумева писање листе инструкција које говоре рачунару шта треба да ради корак по корак како би извршио задатак.

Главне карактеристике процедуралног програмирања јесу:

- **Унапред дефинисане функције:** Унапред дефинисана функција је обично инструкција идентификована именом. Обично су унапред дефинисане функције уграђене у програмске језике вишег нивоа, али се изводе из библиотеке или регистра, а не из програма. Један пример унапред дефинисане функције је „*charAt()*“, која тражи позицију карактера у низу.
- **Локална променљива:** Локална променљива је променљива која је декларисана у главној структури методе и ограничена је на локални досег над којим је дефинисана. Локална променљива може да се користи искључиво у методи у којој је дефинисана, а ако се користи изван дефинисане методе, програм се неће моћи превести или извршити.
- **Глобална променљива:** Глобална променљива је променљива која је декларисана изван сваке друге функције дефинисане у коду. Због тога се глобалне променљиве могу користити у свим функцијама, за разлику од локалне променљиве.
- **Модуларност:** Модуларност представља потребу да два различита система имају два различита задатка, али су груписани како би прво закључили већи задатак. Свака група система тада би имала своје задатке извршене један за другим док сви задаци не буду готови.
- **Пренос параметара:** Пренос параметара је механизам који се користи за прослеђивање параметара функцијама, потпрограмима или процедурама. Пренос параметара може се извршити на два начина „пренос параметара по вредности“, „пренос параметара по референци“.

Проблеми који настају применом процедуралног програмирања јесу ти што постоји претерано коришћење глобалних података: програмери често прибегавају чувању најважнијих података у глобалним променљивама, да би им се лако приступало из свих функција у програму. Међутим, тако се истовремено отварају врата за нехотично уништавање или оштећивање најважнијих података. Такође, пројектовање сложених програма који се тешко одржавају и мењају: иако је препоручљиво да се програм учини веома модуларним, тј. да се издели у велики број логичких функција, постоји граница коју је програмер у стању да сагледа. Реални програми имају на стотине функција чију интеракцију је веома тешко сагледати, па мењање кода постаје изузетно тешко. Пре било какве измене потребно је разумети како ће промена утицати на друге делове програма.

## 2.2. Функционално програмирање

На самом почетку ере рачунара и програмских језика (1940-тих година), програмски језици су развијани у складу за архитектуром рачунара. Један од првих приступа у развоју програма је био императивни јер је концептуално близак хардверу рачунара. Извршавање програма представља извршавање инструкција програма од стране процесора које мењају садржај додељене меморије и ток извршавања програма. Функционално програмирање је настало применом математичких принципа у програмирању. Развојем математичког стила програмирања долази до удаљавања од конкретне рачунарске архитектуре која извршава програм што за последицу има слабије перформансе у односу на језике императивне парадигме. Функционално програмирање посматра појединачне операције као израчунавања математичких функција. Спада у декларативне програмске парадигме јер акценат ставља на то шта посматрани код ради, а не и како ради. Подржава функције које користе само своје улазне параметре тако да позивање функције са истим улазним аргументима увек враћа исти резултат. Овакав приступ омогућава лакше разумевање понашања апликације, елиминацију измена вредности резултата претходних израчунавања (енгл. *mutable data*) и стања апликације односно изазивање бочних ефеката. Програмски језици *C#*, *F#*, *Common LISP*, *Clojure*, *Erlang*, *Haskell*, *Java*, *JavaScript*, *Kotlin* припадају тзв. *impure* групи функционалних језика јер поред функционалног концепта подржавају и објектно-оријентисани и императивни.

Тзв. *pure* функционални језици, нпр. *Clean*, *Curry*, *Elm*, *Mercury*, подржавају само функционално програмирање, али су много мање заступљени у развоју комерцијалних софтверских апликација. Ламбда рачун, настао у 1930-тим година од стране америчког математичара Алонза Черча, представља основу функционалног програмирања. Ламбда рачун уводи неименоване, анонимне, функције које описују неопходне улазне параметре и начин израчунавања резултата.

$\text{square\_num}(x, y) = x^2 + y^2$

$(x, y) \rightarrow x^2 + y^2$

Концепт анонимне функције Ламбда рачун подржава само функције са једним улазним параметром користећи *curing*. Функција са два параметра, може се представити као функција која прима један параметар и враћа нову функцију која, такође, прима један параметар. Помоћу *curing*-а, функција са више параметара се представља као низ функција од којих свака прима само по један параметар. Ламбда рачун посматра функције као вредности прве класе (енгл. *first-class citizen*), те се функције могу користити као улазни параметри или бити враћене као резултат других функција.

*Pure* функцијом се сматра једноставна функција која задовољава следеће критеријуме:

- Прима најмање један улазни параметар
- Приступа вредностима само својих улазних параметара
- Увек враћа резултат
- За исте улазне аргументе увек враћа исти резултат

*Pure* функција користи вредности само својих улазних параметара и не приступа или мења вредност других променљивих. Овим принципом се избегавају непожељни бочни ефекти у програмирању јер утичу на извршавање других функција, уносе логичке грешке у код и отежавају откривање истих. Функција која не прима ниједан улазни параметар и не може да мења вредности променљивих ван свог досега, бескорисна је у програмирању јер може да враћа само константну вредност. Такође, функција мора да врати вредност како би манифестовала своју намену и извршавање.

*Pure* функција за исте улазне параметре увек враћа исти резултат, те је неопходно да и функције које она позива буду, такође, *pure*.

*Pure* функције саме по себи нису довољне за развој модерних софтверских апликација. Апликације подразумевају упис и читање из базе података или фајла, отварање конекције ка удаљеним серверима, те поједине функције раде много више од саме обраде улазних параметара. Извршавање оваквих функција може довести до појаве бочних ефеката или изузетака чиме њихово извршавање постаје непредвидиво и оне постају *impure*. Функционално програмирање не може елиминисати *impure* функције, али тежи да њихову употребу ограничи на она места где су заиста неопходне.

У императивном програмирању, могуће је променљивој променити претходно додељену вредност (*mutable data*). Функционално програмирање подржава концепт непроменљивости који забрањује измену вредности променљивих односно дозвољава само преузимање вредности променљиве. Самим тим, концепт променљиве не постоји, али се ипак назив променљива (енгл. *variable*) користи из историјских разлога. Функционално програмирање чува вредности у константама. Уколико је потребно изменити вредност, креира се нова константа са новом вредношћу. Овим приступом се пише једноставнији и сигурнији код, елиминишу се чак и случајне измене вредности.

Концепт непроменљивости онемогућава употребу *for*, *while* и *do while* петљи јер користе бројачке променљиве и променљиве чијим би се мењањем вредности постигли услови за излазак из петље. Из тих разлога, концепт рекурзије (енгл. *recursion*) је доста заступљен у

функционалном програмирању. У сваком кораку рекурзије, функција враћа конкретну вредност или позива саму себе са новим, израчунатим, вредностима аргумената позива.

За функцију у функционалном програмирању се каже да је *first-class citizen* јер се функција третира као вредности (слично као ниска или цео број). Ово омогућава увођење концепта функција вишег реда (енгл. *higher-order functions*) које могу да примају функцију као аргумент или враћају функцију као повратну вредност. Такође, дефиниција функције се може доделити променљивој или сместити у структуру података. Функционално програмирање подржава и *closure* принцип који повратној анонимној функцији омогућава приступ свим локалним променљивама функције која их је креирала.

*Curring* представља разбијање функције која прима више параметара на композицију више функција од којих свака прима само један параметар.

Применом овог принципа, функција са два параметра се може представити као функција која прима један параметар и враћа другу функцију која прима други параметар. Извршавањем друге функције се добија жељени резултат. Језици који подржавају овај концепт нуде и синтаксу за лакше писање и читање кода.

Функционално програмирање представља сасвим другачији поглед на организацију и извршавање програмског кода у односу на императивни приступ. Већини програмера императивна односно објектно-оријентисана парадигма јесте основна или једина позната пре свега због веће заступљености на рачунарским и техничким факултетима. Разлог овога се може потражити и у мањој примени функционалне парадигме у развоју модерних софтверских апликација пре свега због слабијих перформанси функционалних језика. Применом концепта променљивих и елиминацијом бочних ефеката, функционално програмирање омогућава писање кода са мање логичких грешака, кода у којем се лакше и безбедније могу правити измене. Такође, тестирање и паралелно извршавања функционалног кода је много лакше. Због ових предности, за очекивати је све већу присутност функционалног програмирања како у формалном рачунарском образовању тако и у развоју комерцијалних софтверских решења.

### 2.3. Објектно-оријентисано програмирање

Објектно-оријентирано програмирање је парадигма програмирања, која користи објекте као основу за пројектовање рачунарских програма и различитих апликација софтвера. Заснива се на различитим техникама, као што су енкапсулација, наслеђивање, полиморфизам и апстракција. Решавање проблема парадигмом објектно-оријентисаног програмирања, врло је слично људском начину размишљања и решавању проблема. Састоји се од идентификовања објеката и постављање објеката који ће се користити у одговарајућу секвенцу за решење одређеног проблема. Ради се о дизајну објеката чија ће понашања као јединица и у њиховој међусобној интеракцији, решити одређени проблем.

Интеракција између објеката се састоји у размени порука, где одређена порука усмерена према одређеном објекту, покреће „учаурене“ (енгл. *encapsulated*) операције у том објекту, чиме се решава део обично ширег и сложенијег проблема.

Уопштено гледано, објектно-оријентисано решавање проблема се састоји из четири корака:

1. идентификовање проблема
2. идентификовање објеката који су потребни за његово решење
3. идентификовање порука које ће објекти међусобно слати и примати
4. креирање секвенце порука објектима, које ће решавати проблем или проблеме.

У парадигми објектно-оријентисаног програмирања, објекти су структуре података које представљају одређено и јасно дефинисано знање о спољашњем свету или стварности. Типична организација је у хијерархијске класе, где свака класа објекта поседује информације о особинама објекта које се чувају у инстанцама променљивих и које су повезане (концептом

асоцијације) са сваком инстанцом у одређеној класи. Сваки објекат препознаје други објекат преко његовог интерфејса. Подаци и логика сваког објекта су скривени од других објеката. Тиме се омогућава раздвајање имплементације од понашања објекта у интеракцији са другим објектима.

Структурним програмирањем тежимо да што реалистичније пресликамо (моделујемо, симулирамо) понашање неког система. Објектно-оријентисаним програмирањем пресликавамо исти тај систем тако што га представљамо као скуп међусобно повезаних објеката. Наиме, цели систем се структура на мање целине којима је лакше управљати, а које међусобно комуницирају. Систем који је овако организован лакши је за разумевање, њиме је лакше управљање и рад.

Већина императивних програмских језика (*Fortran, Basic, C*) су само апстракције асемблерског језика па, иако представљају значајан напредак у односу на асемблерски језик, у суштини њихове апстракција леже категорије које одговарају архитектури рачунарског система, а не проблема који се решава. У овом случају, задатак програмера јесте да успостави кореспонденцију измене домена решења (апстрактног модела машине) и домена проблема (апстрактног модела реалног света). Напор који је неопходно уложити у ово пресликавање, а који у једном делу лежи у самој структури коришћеног програмског језика, често нема адекватну вредност будући да резултира програмима ниског степена семантике.

Алтернатива претходно описаном приступу је моделовање проблема који се решава. Први програмски језици који су развијени на проблемској оријентацији уводе специфичне апстракције реалности на којима заснивају поступак пресликавања домена проблема у домен решења.

Типични примери су следећи приступи:

- Све проблеме је могуће апстраховати листом и операцијама над њом (*LISP – List Processing*).
- Сви проблеми су алгоритамске природе (*APL – Algorithmic Programming Language*).
- Сви проблеми се могу исказати као ланци одлучивања (*PROLOG – PROgramming LOGic*).
- Сви проблеми се могу исказати као скуп апстрактних ограничења и манипулација са њима (*Constraint Based Programing*).

Сви наведени приступи представљали су добра решења за одређену уску класу проблема за коју су објективно и дизајнирани, но сваки покушај генерализације искорача ван иницијалног домена готово без изузетка резултује неуспехом. Објектно-оријентисани приступ посебним чини обезбеђење алата механизма за исказивање проблема појмовима из домена проблема. Апстракције које оно уводи су довољно опште тако да програмер није ограничен једним доменом примене.

Објектно-оријентисано програмирање представља имплементациону методу код које су програми организовани као кооперативни скуп објеката при чему сваки објекат представља инстанцу неке класе из хијерархије класа настале на бази релација наслеђивања (енгл. *inheritance*).

Објектно-оријентисани дизајн представља пројектантску методу која обухвата процес објектно-оријентисане декомпозиције и нотацију за исказивање логичких и физичких као и статичких и динамичких модела посматраног система.

Објектно-оријентисана анализа представља аналитичарску методу која захтеве (енгл. *requirements*) посматра из перспективе КЛАСА и ОБЈЕКТА који припада домену проблема. Основни појам у објектно-оријентисаним језицима јесте објекат. Објекат је ентитет који представља људе, места или ствари – било апстрактно било конкретно. Карактеристике

објеката представљене су преко атрибута и метода. Да би се користио готов објекат, није потребно познавати како је он прављен, нити шта је у њему; све што је потребно јесте његова веза са спољним светом – интерфејс, као и то шта објекат ради.

Атрибути описују објекат, вредности атрибута одређују стање објекта. На пример: серијски број, модел, боја и година су само неки од атрибута објекта „Opel“, а једно стање тог објекта би могло да буде *M789876, Kadet*, бела, 2002.

Методе су процедуре додељене објекту које могу да мењају стање објекта (тј. могу да мењају вредности атрибута).

Објекти унутар програма морају да сарађују. Да би омогућили координацију, методе проузрокују и поруке које шаљу другим објектима. Ове поруке могу да активирају методе неких других објеката, а тако активирани методе даље могу да изазову активирање нових акција неког наредног објекта итд. Дакле, поруке омогућавају објекту да комуницира са другим објектима.

Да би неки програмски језик био објектно-оријентисан, он мора да:

- има основну јединицу учуравања (енгл. *encapsulation*) и да подржава „скривање информација“ у оквиру ње (то је класа),
- омогући креирање виших инстанци класе,
- омогући наслеђивање којим се стварају хијерархије класа,
- подржава полиморфизам.

Уколико неки језик нема све ове четири карактеристике, не може се рећи да је објектно оријентисан. На пример, у програмским језицима *Clipper* или *Visual Basic 3.0*, корисник може да користи неке већ предефинисане класе, али не и да ствара нове, те се за ове језике не може рећи да су објектно-оријентисани.

Објектно-оријентисани језици су *SmallTalk*, *Java*, *C++*, *C#*. Наслеђивање (енгл. *inheritance*) је основа развоја објеката; то је систем односно принцип који нам омогућује да креиране ствари не пишемо поново, већ само надоградимо оним што нама треба, за тренутни специфични проблем. Циљ наслеђивања је преузимање особина једног или више родитељских објеката, и њихова доградња на потребан ниво, било додавањем нових метода и података, било применом већ постојећих.

Тако, једном креиран објекат може бити неограничено много пута коришћен чиме се у многоме повећава брзина рада и продуктивност. Наслеђивање представља принцип да једном дефинисана класа, може да произведе одређену класу (или више њих), која наслеђује све атрибуте, методе и поруке те (надређене) класе која ју је генерисала. Подређене класе могу да искључе неке постојеће и укључе нове атрибуте или метода надређене класе. Међутим, то искључивање појединог атрибута или методе није буквално, они и даље формално постоје у тој класи, само им је онемогућен приступ.

#### **2.4. Поређење функционалног и објектно-оријентисаног програмирања**

Објектно-оријентисано програмирање је програмска парадигма, заснована на концепту објекта као градивне јединице апликације, која омогућава дефиницију класе као шаблона (енгл. *template*) за креирање и понашање објеката. Класа дефинише податке (атрибуте) који представљају стање и методе (функције) које представљају понашање објекта као и конструкторе који креирају објекат дате класе. Извршавање објектно-оријентисане апликације се своди на креирање објеката и њихову међусобну интеракцију односно мењање стања објеката.

**Табела 4. Поређење функционалног и објектно-оријентисаног начина програмирања**

<b>Функционално програмирање</b>	<b>Објектно-оријентисано програмирање</b>
Користи непроменљиве податке (захтева више радне меморије)	Користи променљиве податке (штеди радну меморију)
Декларативан приступ	Императиван приступ
Изузетно погодно за паралелно извршавање	Није погодно за паралелно извршавање
Не ствара бочне ефекте	Веома присутни бочни ефекти
Редослед операција није толико битан	Редослед операција је изузетно битан
Користи рекурзију	Користи циклусе (петље)

## 2.5. Поређење објектно-оријентисаног и процедуралног програмирања

Процедурално програмирање ствара корак по корак програм који води апликацију кроз низ наредби. Свака инструкција се изводи по реду. Процедурално програмирање такође се фокусира на идеју да се сви алгоритми извршавају са функцијама и подацима којима програмер има приступ и који је у стању да промени. Објектно оријентисано програмирање је много сличније начину на који функционише стварни свет; аналогна је људском мозгу. Сваки програм састоји се од много ентитета који се називају објекти. Објекти постају основне јединице и с њима су повезане понашања, или специфична сврха. Објекти не могу директно приступити подацима другог објекта. Уместо тога, мора се послати порука која захтева податке, баш као што људи морају тражити информације један од другог; не могу да приступе подацима у мозгу других људи директно.

## ЗАКЉУЧАК

Функционално, процедурално и објектно-оријентисано програмирање представљају различите концепте за програмске језике који их користе. Циљ свих приступа јесте добијање кода без грешака који се могу разумети, лепо кодирати и брзо развијати.

Ови поступци користе различите методе за чување и манипулацију подацима. У функционално програмирању, подаци се не могу чувати у објектима и могу се искључиво трансформисати креирањем функција. У објектно-оријентисано програмирању, подаци се чувају у објектима. Објектно-оријентисано програмирање је распрострањено међу програмерима и користи се веома успешно више од три деценије.

У објектно-оријентисаном програмирању, веома је тешко одржавати објекте повећавањем нивоа у наслеђивању. То такође нарушава принцип енкапсулације, а и смањује модуларнос. У функционалном програмирању, захтева се увек нови објекат како би се извршила функција и то захтева доста меморије при раду апликације.

У сваком случају, на програмерима је да одаберу парадигму која најбоље задовољава њихове потребе и која повећава лакоћу писања, одржавања и проналажење грешака у коду.

## РЕФЕРЕНЦЕ

- Зарић, Ј. (2018). Функционално програмирање и програмски језик F#, Зборник радова Факултета техничких наука, Нови Сад.  
<http://skola.maglaj.net/2/1.%20%20Metodologija%20programiranja.pdf> (19.10.2019.)  
<https://saimaterial.wordpress.com/2007/09/14/1what-is-the-difference-between-object-oriented-programming-and-procedural-programming/> (19.10.2019.)  
[http://wiki.analytica.com/Procedural\\_Programming](http://wiki.analytica.com/Procedural_Programming) (19.10.2019.)  
<https://www.educba.com/functional-programming-vs-oop/> (19.10.2019.)  
<https://hackr.io/blog/procedural-programming> (19.10.2019.)