



Improvement Tracking Language

User guide and syntax manual

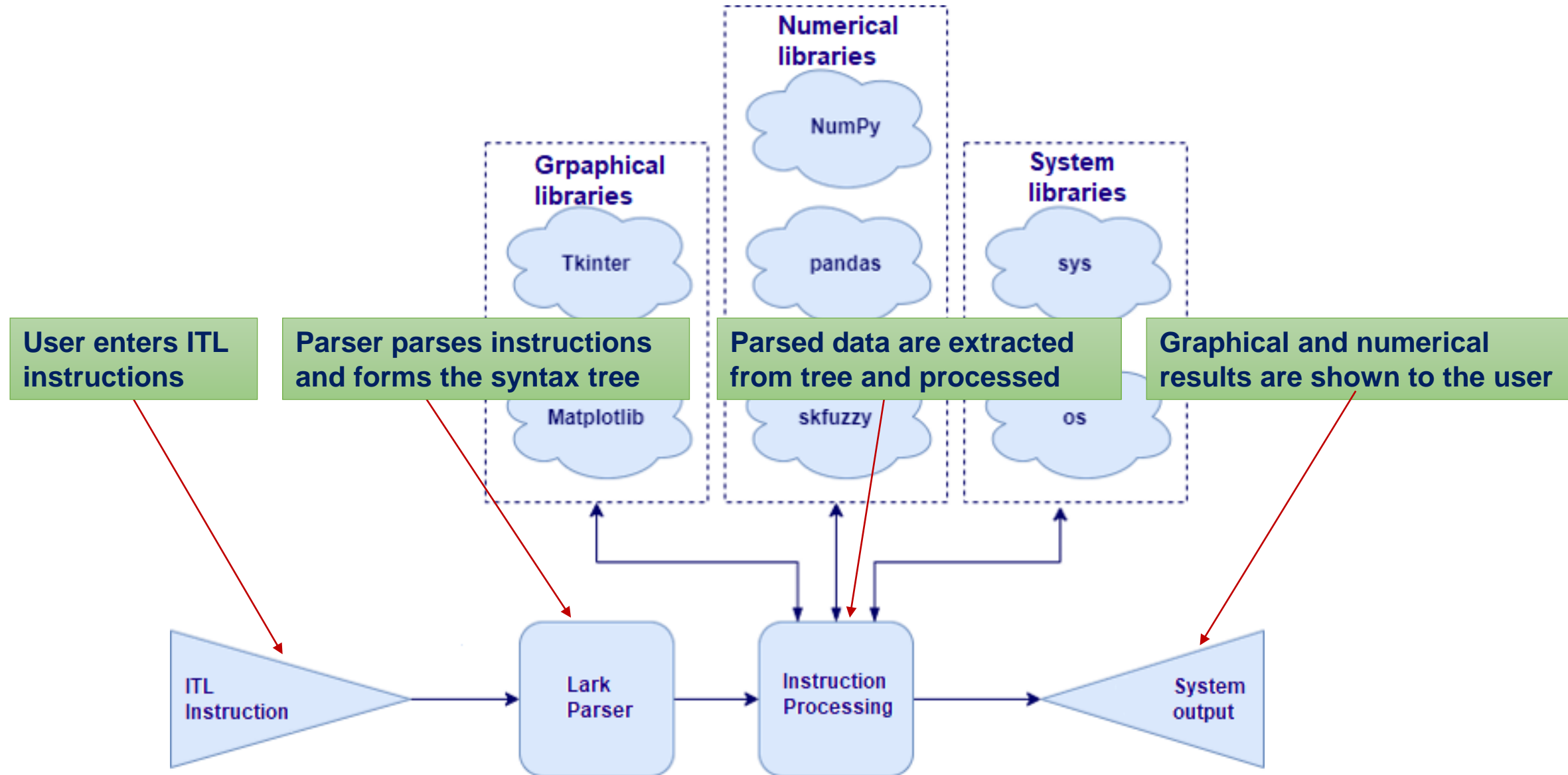
Ver. 1.0

- 1. Intended use and purpose of the ITL**
- 2. ITL Interpreting Block Diagram**
- 3. ITL Syntax**
- 4. ITL Grammar**
- 5. ITL IDE**
- 6. Fuzzy membership functions and performance metrics**
- 7. ITL Case Study – Example**
- 8. ITL Case Study – Example Reporting**
- 9. ITL IDE – Debugging Example**

1. Intended use and purpose of the ITL

- The purpose of the developed language is to provide assessors with an easy way of assessing and visually displaying company performance metric ratings.
- Metrics can be combined from different business domains (for example, metrics for assessing user satisfaction and financial performance metrics for overall assessment of electronic banking web platform).
- For the assessment methodology, the assessment using Fuzzy membership functions was chosen. Upon completing the calculation i.e. assessment process of selected Fuzzy metrics, the visual and numerical results are presented to the assessor. At the end, assessor can investigate assessment results, both by investigating numerical results as far as investigating graphical metrics representation.

2. ITL Interpreting Block Diagram



3. ITL Syntax

izvestaj+ "Naslov izveštaja" → Forms instruction block. Can be defined once or multiple times.

{

metrike+ Skup_metrika_1 → Forms object containing metrics. Can be defined once or multiple times.

{

Metrika_1+ = (v1,b1,a1) → Variable containing numerical values of metric. Can be defined once or multiple times.

}

#komentar → Measured value of metric.

ispisi Skup_metrika_1+; → Instruction for printing data of one or several metrics sets.

oceni Skup_metrika_1; → Instruction for assessing metric inside set of the metrics.

nacrtaj metriku Metrika_1+ iz Skup_metrika_1; → Shows graphical representation of Fuzzy function for one or several metrics inside metrics set. Also shows metrics numerical grade.

oceni uporedno Skup_metrika_1, Skup_metrika_2; → Shows comparative radar chart of the two graded metrics sets.

oceni zbirno Skup_metrika_1+; → Total grade (average value) of one or several metrics sets.

oceni pojedinačno Skup_metrika_1+; → Shows singular grade of the one or several metrics sets.

}

```
graph TD
    A["izvestaj+ \"Naslov izveštaja\""] --> B["Forms instruction block. Can be defined once or multiple times."]
    C["{"] --> D["metrike+ Skup_metrika_1"]
    D --> E["Forms object containing metrics. Can be defined once or multiple times."]
    E --> F["{"]
    F --> G["Metrika_1+ = (v1,b1,a1)"]
    G --> H["Variable containing numerical values of metric. Can be defined once or multiple times."]
    H --> I["v1"]
    H --> J["b1"]
    H --> K["a1"]
    I --> L["Ideal value of measured metric."]
    J --> M["Lowest acceptable value of measured metric."]
    K --> N["Measured value of metric."]
    O["#komentar"] --> N
    P["ispisi Skup_metrika_1+;"] --> Q["Instruction for printing data of one or several metrics sets."]
    R["oceni Skup_metrika_1;"] --> S["Instruction for assessing metric inside set of the metrics."]
    T["nacrtaj metriku Metrika_1+ iz Skup_metrika_1;"] --> U["Shows graphical representation of Fuzzy function for one or several metrics inside metrics set. Also shows metrics numerical grade."]
    V["oceni uporedno Skup_metrika_1, Skup_metrika_2;"] --> W["Shows comparative radar chart of the two graded metrics sets."]
    X["oceni zbirno Skup_metrika_1+;"] --> Y["Total grade (average value) of one or several metrics sets."]
    Z["oceni pojedinačno Skup_metrika_1+;"] --> AA["Shows singular grade of the one or several metrics sets."]
    F --> C2["}"]
    C2 --> C
```

+ represents one or several instructions/variables

4. ITL Grammar

start: instruction+

```
instruction: "izvestaj" STRING code_block -> pocetak_izvestaja
| "metrike" NAME "{" dict_item* "}" -> skup_metrika
| "oceni" NAME ";" -> oceni_metrike
| "oceni zbirno" NAME ("," NAME)* ";" -> oceni_metrike_z
| "oceni uporedno" NAME "," NAME ";" -> oceni_metrike_u
| "oceni pojedinacno" NAME ("," NAME)* ";" -> oceni_metrike_p
| "ispisi" NAME ("," NAME)* ";" -> ispisi_metrike
| "nacrtaj metriku" NAME ("," NAME)* "iz" set -> nacrtaj_metriku
```

code_block: "{" instruction+ "}" -> blok_naredbi

dict_item: NAME "=" dict_subitem -> naziv_metrike

dict_subitem: "(" NUMBER "," NUMBER "," NUMBER ")" -> parametri_metrike

set: NAME ";" -> iz

COMMENT : /#.* /

%import common.CNAME -> NAME

%import common.NUMBER -> NUMBER

%import common.ESCAPED_STRING -> STRING

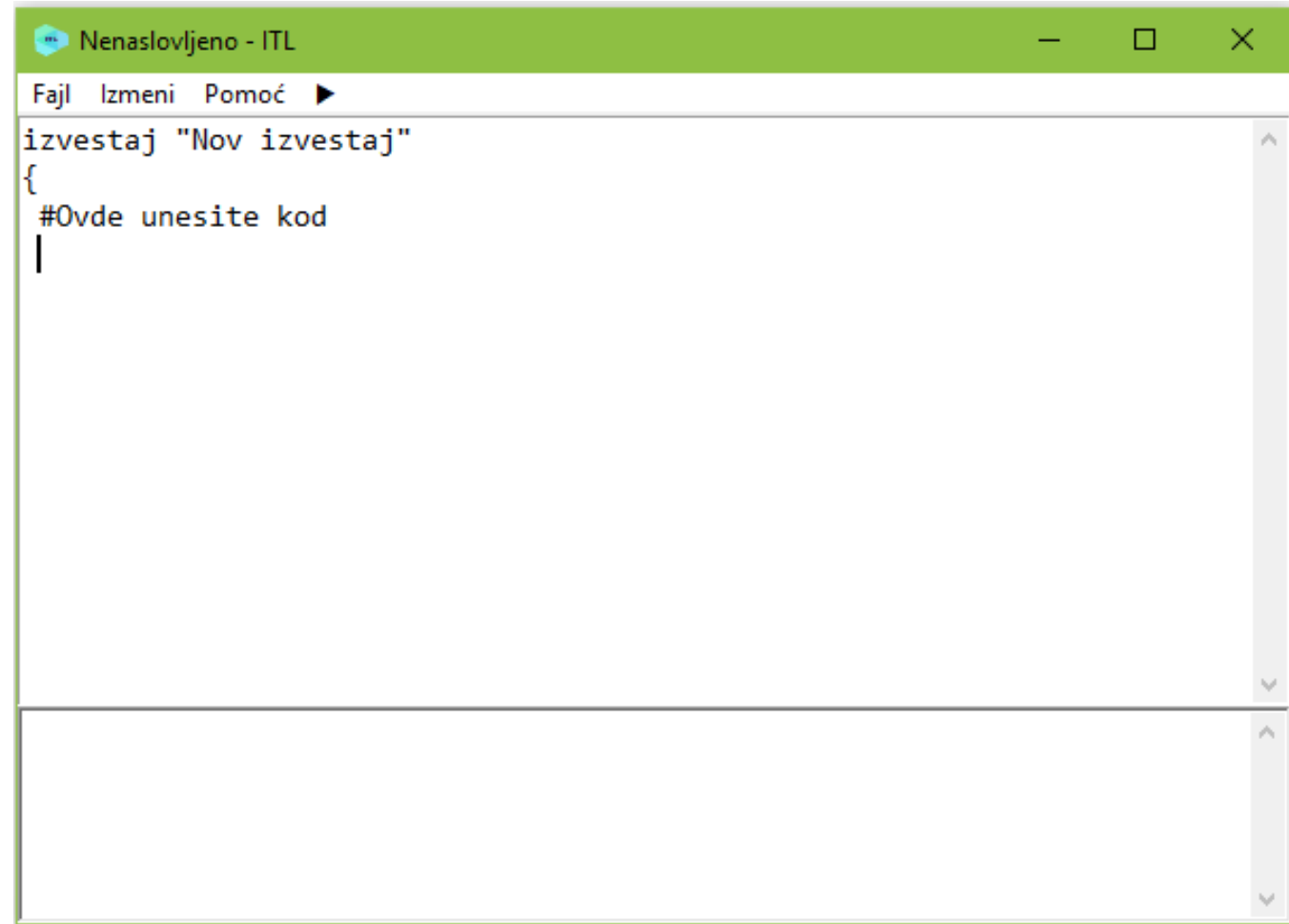
%import common.WS

%ignore WS

%ignore COMMENT

5. ITL IDE

- **Integrated development environment was implemented with the help of the Tkinter Python library.**
- **Supported editor features:**
 - File manipulation (opening, editing, saving)
 - Standard text operations (insert, copy, cut, paste)
 - Undo/Redo operations
 - Code execution
 - Visual support for easy locating errors in syntax
 - Numerical and graphical printing of execution result
 - Automatic populating of editor's text box with code
 - Keyboard shortcuts



6. Fuzzy membership functions and performance metrics

- Rising Fuzzy membership function can be defined as:

$$\mu_{Fi}(v_i) = \begin{cases} 1, & \text{for } v_i \leq a_i \\ 1 - \frac{v_i - a_i}{b - a}, & \text{for } a_i < v_i < b_i \\ 0, & \text{for } v_i \geq b_i \end{cases}$$

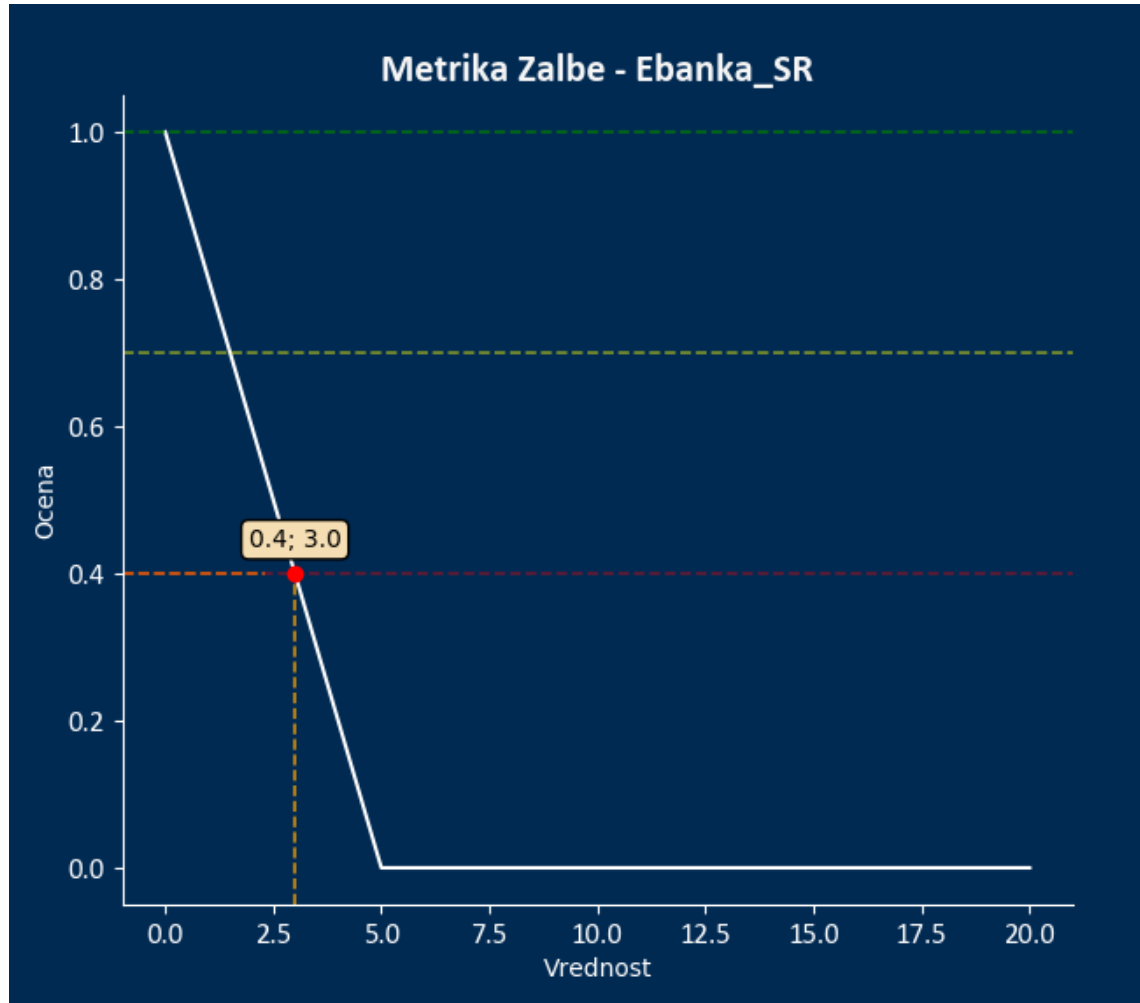
where a_i , b_i , v_i , are best, worst and measured value of i –th metric respectively and, $\mu_{Fi}(v_i)$ representing i –th metric's grade between 0 and 1

- Falling Fuzzy membership function can be defined as:

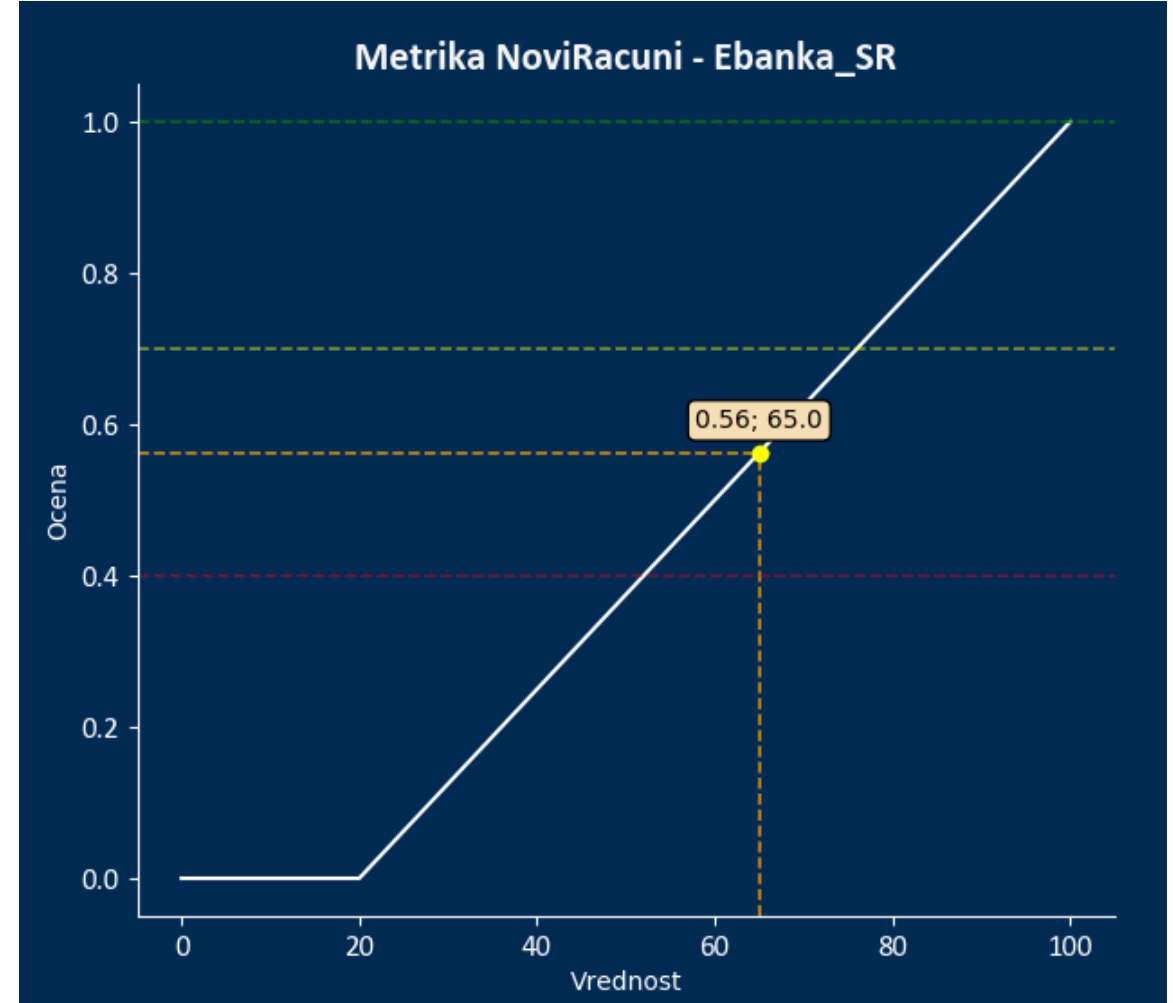
$$\mu_{Fi}(v_i) = \begin{cases} 0, & \text{for } v_i \leq a_i \\ \frac{v_i - a_i}{b - a}, & \text{for } a_i < v_i < b_i \\ 1, & \text{for } v_i \geq b_i \end{cases}$$

where a_i , b_i , v_i , are best, worst and measured value of i –th metric respectively and, $\mu_{Fi}(v_i)$ representing i –th metric's grade between 0 and 1

6. Fuzzy membership functions and performance metrics



Falling Fuzzy membership function



Rising Fuzzy membership function

7. ITL Case Study - Example

- We will define five metrics that will be evaluated through this evaluation system. Metric scores are calculated using the Python Skfuzzy Library.
- We will measure the quality of user experience for three imaginary e-banking systems of three banks within the same group of banks.
- Metrics definitions (monitored on a monthly basis):

1. Complaints

Definition: Number of users complaints on web application performance.

Metric type: Falling Fuzzy memb. func.

Ideal value: 0 (b=0)

Acceptable value: 5 (a=5)

2. Time to loan approval

Definition: Average time passed from sending a loan application until loan approval

Metric type: Falling Fuzzy memb. func.

Ideal value: 0 (b=0)

Acceptable value: 15 days (a=15)

3. Amount of approved loans

Definition: Amount of approved loans

Metric type: Rising Fuzzy memb. func.

Ideal value: 500 (b=500)

Acceptable value: 30 (a=30)

4. Application screen time

Definition: Average time spent using web application (in one day)

Metric type: Rising Fuzzy memb. func.

Ideal value: 15 minutes (b=15)

Acceptable value: 1 minute (a=1)

5. New issued credit cards

Definition: Amount of new issued credit cards

Metric type: Rising Fuzzy memb. func.

Ideal value: 400 pcs. (b=400)

Acceptable value: 100 pcs. (a=100)

7. ITL Case Study - Example

```
izvestaj "Nedeljni izvestaj - Grupa E-banka"  
{
```

We define metrics sets for all three banks

```
metrike E_bank_1 {  
    Zalbe = (3, 20, 5) # Defining metric Zalbe  
    NoveKredKartice = (218, 100, 400)  
    ProvedenoVreme = (7.4, 1, 15)  
    NovKredit = (305, 30, 500)  
    VremeOdobrenjaKredita = (4.43, 30, 15)  
}
```

```
metrike E_bank_2 {  
    Zalbe = (1, 20, 5)  
    NoveKredKartice = (295, 100, 400)  
    ProvedenoVreme = (4.9, 1, 15)  
    NovKredit = (352, 30, 500)  
    VremeOdobrenjaKredita = (8.82, 30, 15)  
}
```

```
metrike E_bank_3 {  
    Zalbe = (4, 20, 5)  
    NoveKredKartice = (191, 100, 400)  
    ProvedenoVreme = (9.5, 1, 15)  
    NovKredit = (254, 30, 500)  
    VremeOdobrenjaKredita = (2.52, 30, 15)  
}
```

7. ITL Case Study – Example

Print out overall grade of all three banks

oceni zbirno E_bank_1, E_bank_2, E_bank_3;

Assess individually grades of each bank

oceni pojedinacno E_bank_1, E_bank_2, E_bank_3;

Third bank got worst grade. Let's examine why:

oceni E_bank_3;

Third bank made worst result for metrics Zalbe and NoveKredKartice.

Let's draw them so we can better examine them:

nacrtaj metriku Zalbe, NoveKredKartice iz E_bank_3;

At the end, we compare overall successfulness of first and second bank:

oceni uporedno E_bank_1, E_bank_2;

To print all metrics data (entered raw data) we can use the following instruction:

ispisi E_bank_1, E_bank_2, E_bank_3;

}

8. ITL Case Study – Example Reporting

Example of assessment grade for E_bank_3 set of metrics

Ocena metrike:

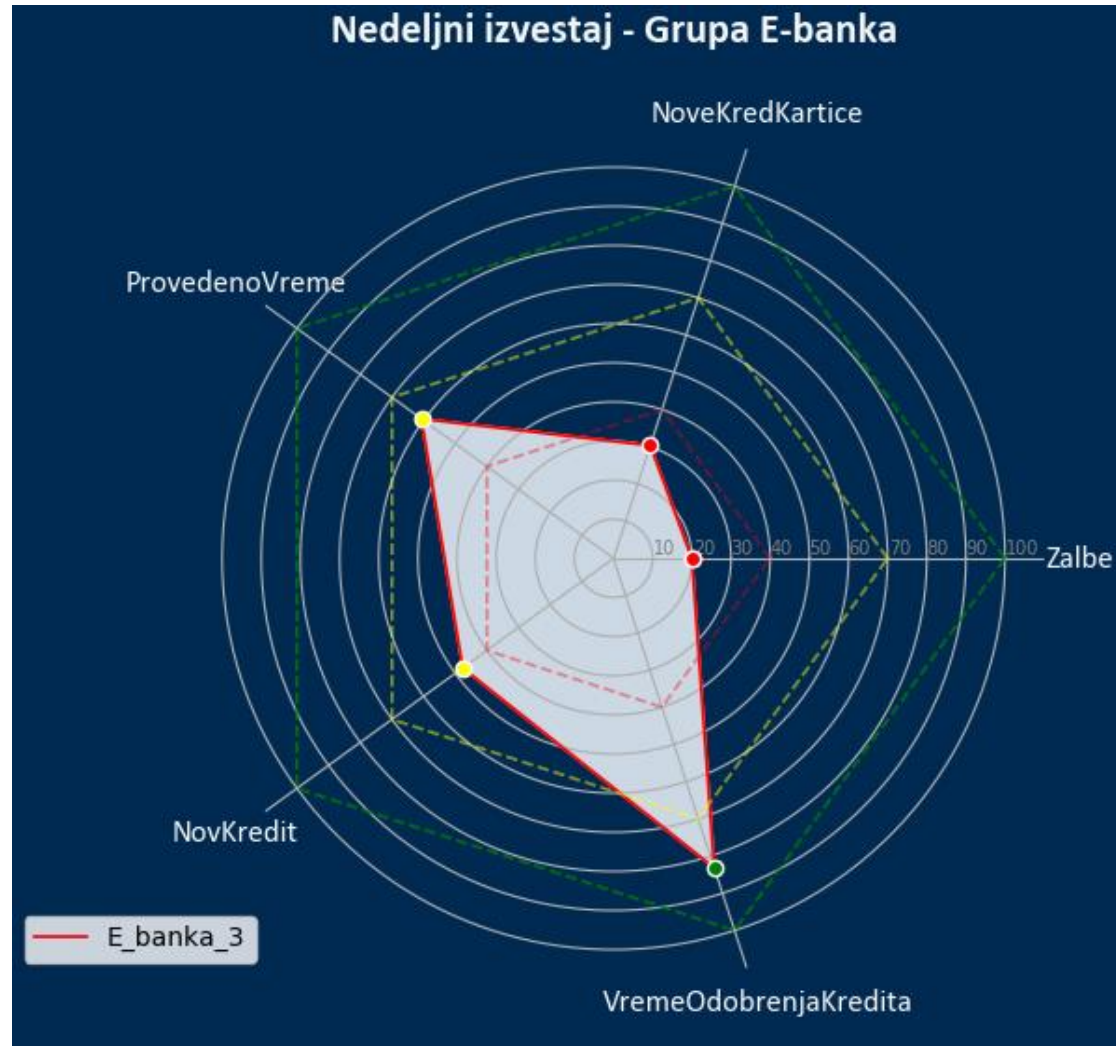
Metrika: {'E_bank_3': {'Zalbe': 20.0, 'NoveKredKartice': 30.333, 'ProvedenoVreme': 60.714, 'NovKredit': 47.66, 'VremeOdobren: Kredita': 83.2}}

Ocena: 48/100 bodova

Crtanje metrika: E_bank_3

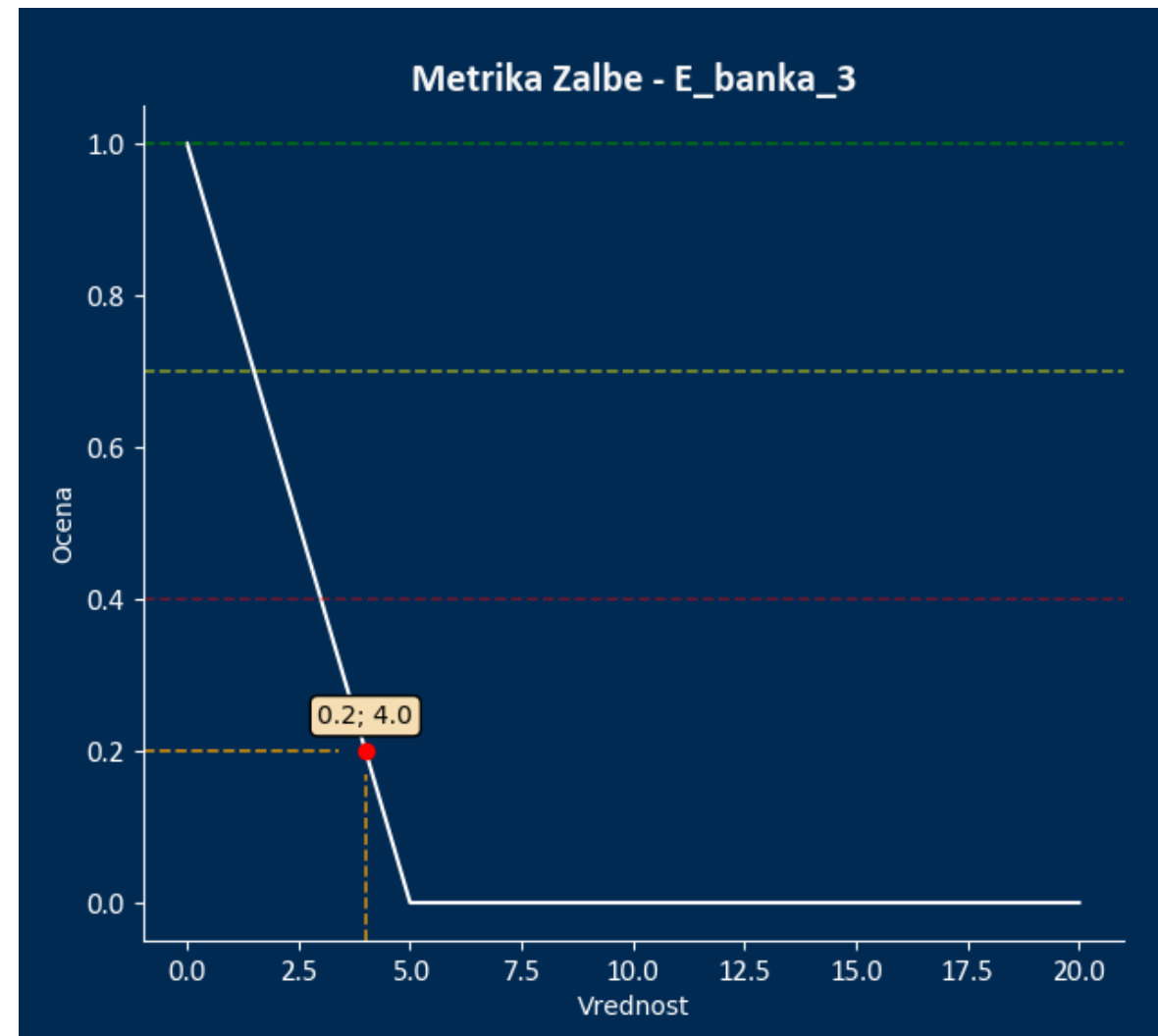
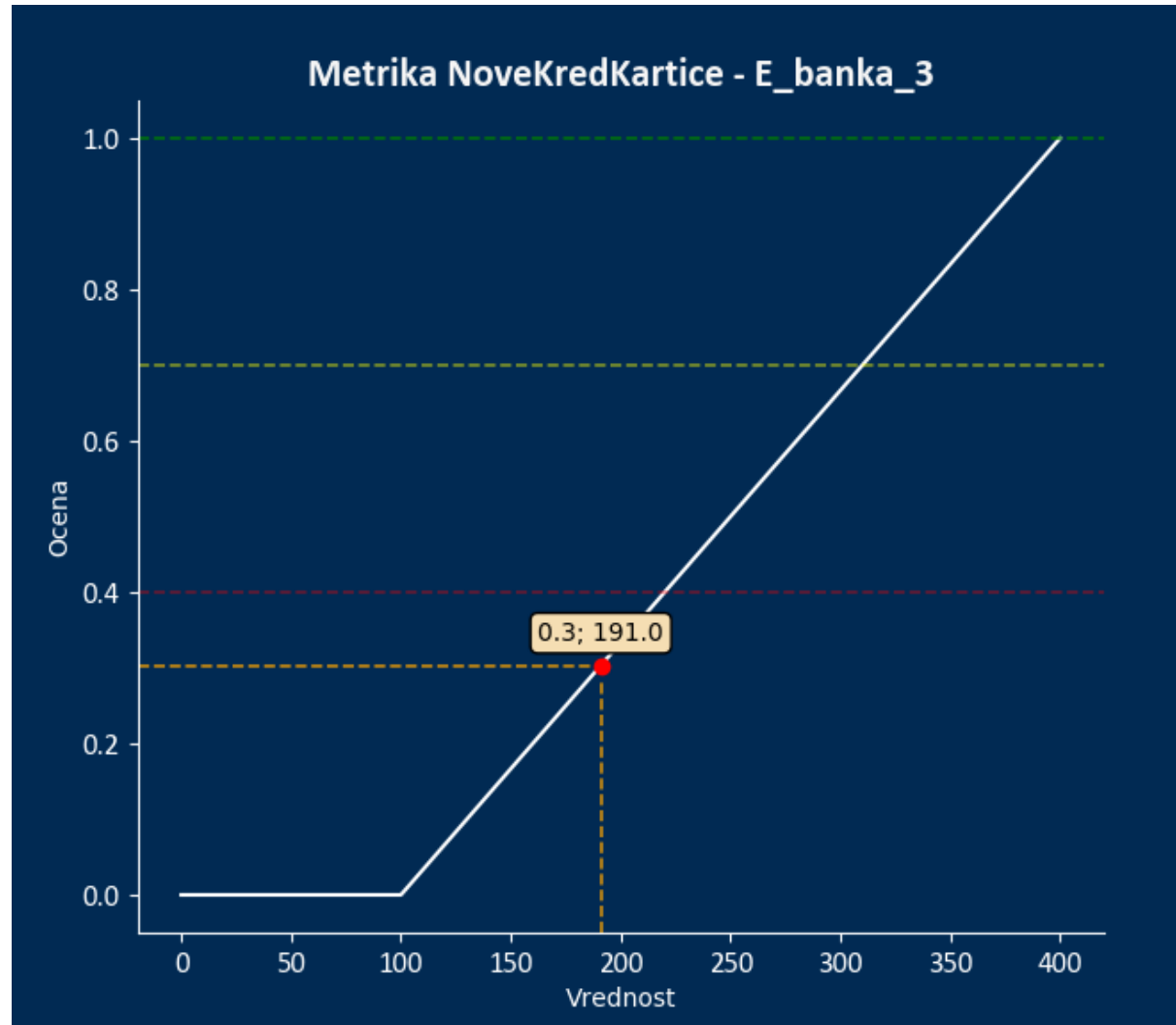
8. ITL Case Study – Example Reporting

Grade radar chart for E_banka_3 metrics set



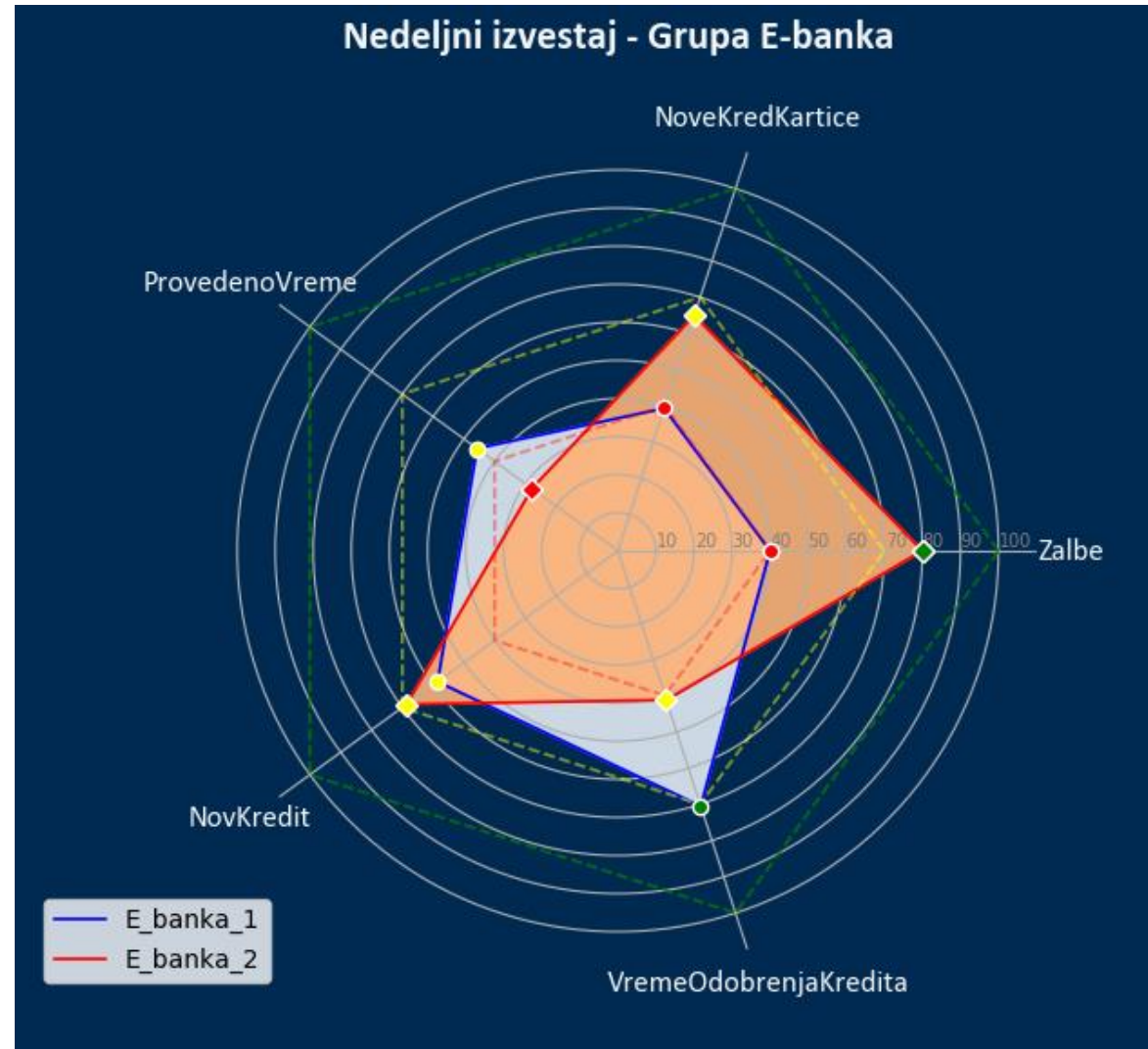
8. ITL Case Study – Example Reporting

Fuzzy metrics with corresponding grades for two metrics of E_bank_a_3

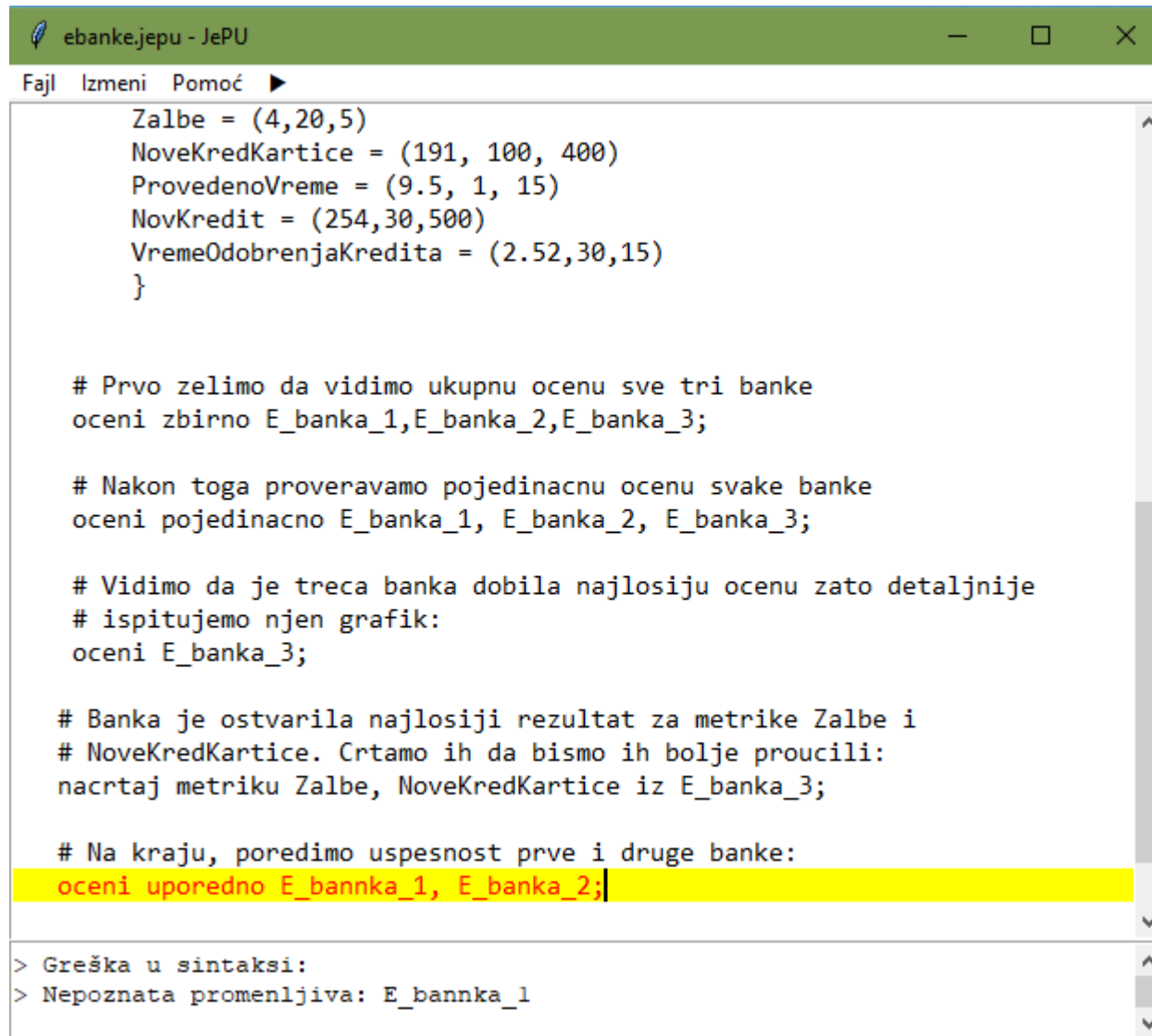


8. ITL Case Study – Example Reporting

Comparative radar chart of grades for E_bank_1 and E_bank_2



9. ITL IDE – Debugging Example



```
ebanke.jepu - JePU
Fajl  Izmeni  Pomoć ►

Zalbe = (4,20,5)
NoveKredKartice = (191, 100, 400)
ProvedenoVreme = (9.5, 1, 15)
NovKredit = (254,30,500)
VremeOdobrenjaKredita = (2.52,30,15)
}

# Prvo zelimo da vidimo ukupnu ocenu sve tri banke
oceni zbirno E_bank_1,E_bank_2,E_bank_3;

# Nakon toga proveravamo poedinacnu ocenu svake banke
oceni poedinacno E_bank_1, E_bank_2, E_bank_3;

# Vidimo da je treca banka dobila najlosiju ocenu zato detaljnije
# ispitujemo njen grafik:
oceni E_bank_3;

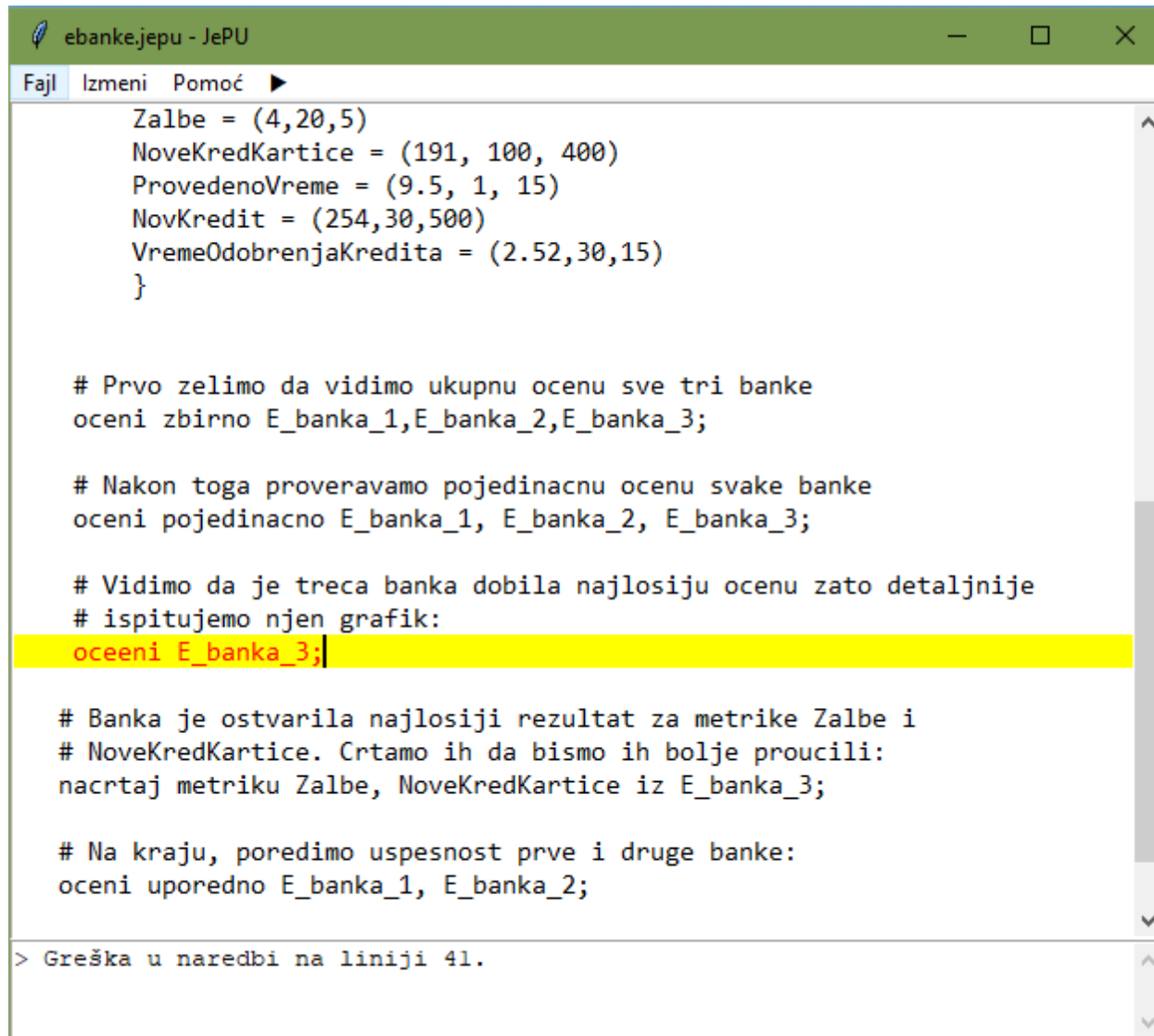
# Banka je ostvarila najlosiji rezultat za metrike Zalbe i
# NoveKredKartice. Crtamo ih da bismo ih bolje proucili:
nacrtaj metriku Zalbe, NoveKredKartice iz E_bank_3;

# Na kraju, poredimo uspesnost prve i druge banke:
oceni uporedno E_bannka_1, E_bank_2;

> Greška u sintaksi:
> Nepoznata promenljiva: E_bannka_1
```

- Example of error in syntax.
Misspelled variable name E_bank_1 (E_bannka_1)
- IDE visually marks the error and prints its contents in terminal.

9. ITL IDE – Debugging Example



```
eбанке.jepu - JePU
Fajl  Izmeni  Pomoć ►

Zalbe = (4,20,5)
NoveKredKartice = (191, 100, 400)
ProvedenoVreme = (9.5, 1, 15)
NovKredit = (254,30,500)
VremeOdobrenjaKredita = (2.52,30,15)
}

# Prvo zelimo da vidimo ukupnu ocenu sve tri banke
oceni zbirno E_bank_1,E_bank_2,E_bank_3;

# Nakon toga proveravamo pojedinačnu ocenu svake banke
oceni pojedinačno E_bank_1, E_bank_2, E_bank_3;

# Vidimo da je treća banka dobila najlosiju ocenu zato detaljnije
# ispitujemo njen grafik:
oceeni E_bank_3;

# Banka je ostvarila najlosiji rezultat za metrike Zalbe i
# NoveKredKartice. Crtamo ih da bismo ih bolje proučili:
nacrtaj metriku Zalbe, NoveKredKartice iz E_bank_3;

# Na kraju, poredimo uspesnost prve i druge banke:
oceni uporedno E_bank_1, E_bank_2;

> Greška u naredbi na liniji 41.
```

- Example of error in syntax.
Misspelled instruction name **oceni** (**ocenni**)
- IDE visually marks the error and prints its contents in terminal with error line number.



Improvement Tracking Language

User guide and syntax manual

Ver. 1.0