

# PROGRAMSKA POTPORA KOMUNIKACIJSKIM SUSTAVIMA

## Prva laboratorijska vježba

### Klasa *Professional*

Klasa *Professional* roditeljska je klasa koja služi kao temelj za stvaranje objekata koji predstavljaju osobe u različitim profesionalnim područjima poput umjetnika, znanstvenika i poduzetnika.

#### Atributi klase *Professional*:

- *first\_name* (*string*): ime osobe
- *last\_name* (*string*): prezime osobe
- *age* (*string*): dob osobe.
- *expertise\_level* (*int*): razina stručnosti osobe
- *salary* (*float*): plaća osobe
- *nationality* (*string*): nacionalnost osobe

#### Metode klase *Professional*:

- *\_\_init\_\_(self, first\_name, last\_name, age, expertise\_level, salary, nationality)*: konstruktor klase *Professional*
- *name(self)* -> *str*: vraća puno ime osobe.
- *personal\_info(self)* -> *str*: vraća osnovne informacije o osobi (dob, nacionalnost, specijalnost)
- *achievements(self)* -> *str*: vraća postignuća osobe
- *calculate\_avg\_salary(professionals: List[Professional])* -> *float*: statička metoda koja računa prosječnu plaću osobe
- *rank\_by\_salary(professionals: List[Professional])* -> *List[Professional]*: statička metoda koja rangira (sortira) osobe silazno prema plaći
- *perform\_action(self)* -> *str*: vraća radnju koju osoba izvodi

## Klasa *Artist*

Klasa *Artist* nasljeđuje klasu *Professional* i predstavlja osobu koja je bavi umjetnošću.

Atributi klase *Artist*:

- svi atributi iz klase *Professional*
- *art\_style (string)*: umjetnički stil kojim se umjetnik bavi
- *awards (int)*: broj nagrada koje je umjetnik dobio

Metode klase *Artist*:

- *\_\_init\_\_(self, first\_name, last\_name, age, expertise\_level, salary, nationality, art\_style, awards)*: konstruktor klase *Artist*
- *personal\_info(self)* -> str: **polimorfizam**; vraća osnovne informacije o umjetniku s detaljem vezanim za umjetnike (atribut *art\_style*)
- *achievements(self)* -> str: **polimorfizam**; vraća postignuća umjetnika, specifična za umjetnike
- *create\_masterpiece(self)* -> str: vraća poruku o stvaranju remek-djela
- *practice(self)* -> str: vraća poruku o tome kako umjetnik provodi vrijeme vježbajući
- *perform\_action(self)* -> str: **polimorfizam**; vraća radnju koju umjetnik izvodi, specifičnu za umjetnike

## Klasa Scientist

Klasa *Scientist* također nasljeđuje klasu *Professional* i predstavlja osobu koja se bavi znanosti.

Atributi klase *Scientist*:

- svi atributi klase *Professional*
- *field* (string): područje istraživanja znanstvenika
- *publications* (int): broj radova koje je znanstvenik objavio

Metode klase *Scientist*:

- `__init__(self, first_name, last_name, age, expertise_level, salary, nationality, field, publications)`: konstruktor klase *Scientist*
- `personal_info(self)` -> str: **polimorfizam**; vraća osnovne informacije o znanstveniku s detaljem vezanim za znanstvenike (atribut *field*)
- `achievements(self)` -> str: **polimorfizam**; vraća postignuća znanstvenika, specifična za znanstvenike
- `conduct_experiment(self)` -> str: vraća poruku o provođenju eksperimenata
- `analyze_data(self)` -> str: vraća poruku o analizi podataka
- `perform_action(self)` -> str: **polimorfizam**; vraća radnju koju znanstvenik izvodi, specifičnu za znanstvenike

### Klasa *Entrepreneur*

Klasa *Entrepreneur* također nasljeđuje klasu *Professional* i predstavlja osobu koja se bavi poduzetništvom.

Atributi klase *Entrepreneur*:

- svi atributi klase *Professional*
- *industry* (*string*): industrija u kojoj poduzetnik djeluje
- *ventures* (*int*): broj poduzetničkih pothvata koje je poduzetnik poduzeo

Metode klase *Entrepreneur*:

- *\_\_init\_\_(self, first\_name, last\_name, age, expertise\_level, salary, nationality, industry, ventures)*: konstruktor klase *Entrepreneur*
- *personal\_info(self)* -> *str*: **polimorfizam**; vraća osnovne informacije o poduzetniku s detaljem vezanim za poduzetnike (atribut *industry*)
- *achievements(self)* -> *str*: **polimorfizam**; vraća postignuća poduzetnika, specifična za poduzetnike
- *launch\_startup(self)* -> *str*: vraća poruku o pokretanju startupa
- *negotiate\_deals(self)* -> *str*: vraća poruku o pregovaranju oko poslova
- *perform\_action(self)* -> *str*: **polimorfizam**; vraća radnju koju poduzetnik izvodi, specifičnu za poduzetnike

## Principi objektno orijentiranog programiranja (nasljeđivanje i polimorfizam):

### 1. Nasljeđivanje

Klasa *Artist* nasljeđuje klasu *Professional*, što omogućuje umjetnicima pristup zajedničkim atributima i metodama „profesionalnim“ osobama općenito, ali i dodavanje specifičnih atributa i metoda za umjetnike.

```
class Artist(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, art_style, awards):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.art_style = art_style
        self.awards = awards
```

Klasa *Scientist* nasljeđuje klasu *Professional*, što omogućuje znanstvenicima pristup zajedničkim atributima i metodama "profesionalnim" osobama općenito, ali i dodavanje specifičnih atributa i metoda za znanstvenike.

```
class Scientist(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, field, publications):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.field = field
        self.publications = publications
```

Klasa *Entrepreneur* nasljeđuje klasu *Professional*, što omogućuje poduzetnicima pristup zajedničkim atributima i metodama "profesionalnim" osobama općenito, ali i dodavanje specifičnih atributa i metoda za poduzetnike.

```
class Entrepreneur(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, industry, ventures):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.industry = industry
        self.ventures = ventures
```

## 2. Polimorfizam

Metode *personal\_info()*, *achievements()* i *perform\_action()* polimorfne su u sve tri klase koje nasljeđuju roditeljsku klasu *Professional*: *Artist*, *Scientist* i *Entrepreneur*, pružajući različito ponašanje ovisno o tipu objekta koji ih poziva.

U metodi *personal\_info()* razlika je u ispisu jednog atributa (*art\_style/field/industry*) ovisno o tipu objekta koji je poziva.

U metodi *achievements()* razlika je u razini stručnosti (*enterprise\_level*) koja se koristi za provjeru u uvjetima metode, a to također ovisi o tipu objekta koji ovu metodu poziva.

U metodi *perform\_action()* o vrsti radnji koju određena osoba izvodi, a to kao i za gornje dvije polimorfne metode ovisi o vrsti objekta koji ovu metodu poziva.

Primjer polimorfizma između roditeljske klase *Professional* i klase *Artists* koja ju nasljeđuje (cijeli se kod može vidjeti na kraju ovoga dokumenta, kao i svi polimorfizmi):

```
class Professional:
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality):
        self.first_name = first_name
        self.last_name = last_name
        self.age = str(age)
        self.expertise_level = expertise_level
        self.salary = salary
        self.nationality = nationality

    def name(self):
        return str(self.first_name + " " + self.last_name)

    def personal_info(self):
        return "Personal information about an artist (age, nationality, specialty) ..."

    def achievements(self):
        return "accomplished an achievement."

    def perform_action(self):
        return "performs a professional action."

class Artist(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, art_style, awards):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.art_style = art_style
        self.awards = awards

    # Polimorfizam
    def personal_info(self):
        return str("Personal information about " + self.name() + " (age, nationality, specialty): "
                  + self.age + ", " + self.nationality + ", " + self.art_style + " art")

    # Polimorfizam
    def achievements(self):
        # enterprise_level ∈ [1, 10]
        if self.expertise_level < 4:
            return f"{self.name()} is still exploring their potential in the world of {self.art_style} art."
        elif 4 <= self.expertise_level < 8:
            return f"{self.name()} has established a unique style in {self.art_style} art and won {self.awards} awards."
        else:
            return f"{self.name()} is a master of {self.art_style} art, with {self.awards} prestigious awards to their name."

    # Polimorfizam
    def perform_action(self):
        actions = ["creates a masterpiece", "practices their art", "showcases their work"]
        action = random.choice(actions)
        return f"{self.name()} {action}."
```

## Prikaz koda skripte *professional.py*:

```
import random

class Professional:
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality):
        self.first_name = first_name
        self.last_name = last_name
        self.age = str(age)
        self.expertise_level = expertise_level
        self.salary = salary
        self.nationality = nationality

    def name(self):
        return str(self.first_name + " " + self.last_name)

    def personal_info(self):
        return str("Personal information about " + self.name() + " (age, nationality, specialty): "
                  + self.age + ", " + self.nationality + ", " + "It's necessary to specify the profession!")

    def achievements(self):
        return f"{self.name()} accomplished an achievement. For more details please specify the profession!"

    @staticmethod
    def calculate_avg_salary(professionals):
        total_salary = sum([prof.salary for prof in professionals])
        return total_salary / len(professionals)

    @staticmethod
    def rank_by_salary(professionals):
        ranked_professionals = sorted(professionals, key=lambda x: x.salary, reverse=True)
        return ranked_professionals

    def perform_action(self):
        return f"{self.name()} performs a professional action. For more details please specify the profession!"

class Artist(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, art_style, awards):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.art_style = art_style
        self.awards = awards

    # Polimorfizam
    def personal_info(self):
        return str("Personal information about " + self.name() + " (age, nationality, specialty): "
                  + self.age + ", " + self.nationality + ", " + self.art_style + " art")

    # Polimorfizam
    def achievements(self):
        # enterprise_level ∈ [1, 10]
        if self.expertise_level < 4:
            return f"{self.name()} is still exploring their potential in the world of {self.art_style} art."
        elif 4 <= self.expertise_level < 8:
            return f"{self.name()} has established a unique style in {self.art_style} art and won {self.awards} awards."
        else:
            return f"{self.name()} is a master of {self.art_style} art, with {self.awards} prestigious awards to their name."

    def create_masterpiece(self):
        masterpiece_types = ["painting", "sculpture", "music composition", "poetry"]
        masterpiece = random.choice(masterpiece_types)
        return f"{self.name()} creates a masterpiece {masterpiece}!"

    def practice(self):
        return f"{self.name()} spends hours practicing their art."

    # Polimorfizam
    def perform_action(self):
        actions = ["creates a masterpiece", "practices their art", "showcases their work"]
        action = random.choice(actions)
        return f"{self.name()} {action}."
```

## Prikaz koda skripte *professional.py* (nastavak):

```
class Scientist(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, field, publications):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.field = field
        self.publications = publications

    # Polimorfizam
    def personal_info(self):
        return str("Personal information about " + self.name() + " (age, nationality, specialty): "
                  + self.age + ", " + self.nationality + ", " + self.field + " research")

    # Polimorfizam
    def achievements(self):
        # enterprise_level ∈ [1, 5]
        if self.expertise_level < 2:
            return f"{self.name()} is still grasping the fundamentals of {self.field} research."
        elif 2 <= self.expertise_level < 4:
            return f"{self.name()} has made significant breakthroughs in {self.field} and published {self.publications} papers."
        else:
            return f"{self.name()} is a leading authority in {self.field}, with {self.publications} influential publications to their name."

    def conduct_experiment(self):
        return f"{self.name()} conducts a groundbreaking experiment in {self.field}!"

    def analyze_data(self):
        return f"{self.name()} spends days analyzing the collected data."

    # Polimorfizam
    def perform_action(self):
        actions = ["conducts an experiment", "analyzes data", "publishes a paper"]
        action = random.choice(actions)
        return f"{self.name()} {action}."

class Entrepreneur(Professional):
    def __init__(self, first_name, last_name, age, expertise_level, salary, nationality, industry, ventures):
        super().__init__(first_name, last_name, age, expertise_level, salary, nationality)
        self.industry = industry
        self.ventures = ventures

    # Polimorfizam
    def personal_info(self):
        return str("Personal information about " + self.name() + " (age, nationality, specialty): "
                  + self.age + ", " + self.nationality + ", " + self.industry + " industry")

    # Polimorfizam
    def achievements(self):
        # enterprise_level ∈ [0, 3]
        if self.expertise_level < 1:
            return f"{self.name()} is navigating the challenges of the {self.industry} industry."
        elif 1 <= self.expertise_level <= 2:
            return f"{self.name()} has successfully launched and sustained {self.ventures} ventures in the {self.industry} sector."
        else:
            return f"{self.name()} is a titan of the {self.industry} industry, with {self.ventures} successful ventures reshaping global markets."

    def launch_startup(self):
        return f"{self.name()} successfully launches a startup in the {self.industry} industry!"

    def negotiate_deals(self):
        return f"{self.name()} negotiates lucrative deals for their company."

    # Polimorfizam
    def perform_action(self):
        actions = ["launches a startup", "negotiates deals", "networks"]
        action = random.choice(actions)
        return f"{self.name()} {action}."
```



Ispis nakon pokretanja skripte *main.py* (skripta nije prikazana u ovom dokumentu, može se pronaći u predanoj *.zip* mapi):

- u skripti *main.py* može se vidjeti inicijalizacija traženih pet objekata za svaku klasu i pozivi svih metoda u klasama

```
Georgia O'Keeffe
Georgia O'Keeffe accomplished an achievement. For more details please specify the profession!
Personal information about Georgia O'Keeffe (age, nationality, specialty): 89, American, It's necessary to specify the profession!
Georgia O'Keeffe performs a professional action. For more details please specify the profession!

Isaac Newton
Isaac Newton accomplished an achievement. For more details please specify the profession!
Personal information about Isaac Newton (age, nationality, specialty): 64, English, It's necessary to specify the profession!
Isaac Newton performs a professional action. For more details please specify the profession!

Steve Jobs
Steve Jobs accomplished an achievement. For more details please specify the profession!
Personal information about Steve Jobs (age, nationality, specialty): 56, American, It's necessary to specify the profession!
Steve Jobs performs a professional action. For more details please specify the profession!

Martha Stewart
Martha Stewart accomplished an achievement. For more details please specify the profession!
Personal information about Martha Stewart (age, nationality, specialty): 80, American, It's necessary to specify the profession!
Martha Stewart performs a professional action. For more details please specify the profession!

Leonhard Euler
Leonhard Euler accomplished an achievement. For more details please specify the profession!
Personal information about Leonhard Euler (age, nationality, specialty): 76, Swiss, It's necessary to specify the profession!
Leonhard Euler performs a professional action. For more details please specify the profession!

Leonardo da Vinci is a master of Renaissance art, with 5 prestigious awards to their name.
Personal information about Leonardo da Vinci (age, nationality, specialty): 28, Italian, Renaissance art
Leonardo da Vinci creates a masterpiece sculpture!
Leonardo da Vinci spends hours practicing their art.
Leonardo da Vinci showcases their work.

Vincent van Gogh is a master of Post-Impressionist art, with 3 prestigious awards to their name.
Personal information about Vincent van Gogh (age, nationality, specialty): 45, Dutch, Post-Impressionist art
Vincent van Gogh creates a masterpiece poetry!
Vincent van Gogh spends hours practicing their art.
Vincent van Gogh showcases their work.

Pablo Picasso is a master of Cubism art, with 7 prestigious awards to their name.
Personal information about Pablo Picasso (age, nationality, specialty): 33, Spanish, Cubism art
Pablo Picasso creates a masterpiece poetry!
Pablo Picasso spends hours practicing their art.
Pablo Picasso creates a masterpiece.

Frida Kahlo is a master of Surrealism art, with 4 prestigious awards to their name.
Personal information about Frida Kahlo (age, nationality, specialty): 40, Mexican, Surrealism art
Frida Kahlo creates a masterpiece painting!
Frida Kahlo spends hours practicing their art.
Frida Kahlo creates a masterpiece.

Jackson Pollock is a master of Abstract Expressionism art, with 6 prestigious awards to their name.
Personal information about Jackson Pollock (age, nationality, specialty): 50, American, Abstract Expressionism art
Jackson Pollock creates a masterpiece music composition!
Jackson Pollock spends hours practicing their art.
Jackson Pollock showcases their work.

Marie Curie has made significant breakthroughs in Physics and published 10 papers.
Personal information about Marie Curie (age, nationality, specialty): 35, Polish, Physics research
Marie Curie conducts a groundbreaking experiment in Physics!
Marie Curie spends days analyzing the collected data.
Marie Curie conducts an experiment.

Albert Einstein has made significant breakthroughs in Theoretical Physics and published 15 papers.
Personal information about Albert Einstein (age, nationality, specialty): 50, German, Theoretical Physics research
Albert Einstein conducts a groundbreaking experiment in Theoretical Physics!
Albert Einstein spends days analyzing the collected data.
Albert Einstein publishes a paper.
```

## Ispis nakon pokretanja skripte *main.py* (nastavak):

Stephen Hawking has made significant breakthroughs in Cosmology and published 12 papers.  
Personal information about Stephen Hawking (age, nationality, specialty): 40, British, Cosmology research  
Stephen Hawking conducts a groundbreaking experiment in Cosmology!  
Stephen Hawking spends days analyzing the collected data.  
Stephen Hawking publishes a paper.

Ada Lovelace has made significant breakthroughs in Computer Science and published 8 papers.  
Personal information about Ada Lovelace (age, nationality, specialty): 38, British, Computer Science research  
Ada Lovelace conducts a groundbreaking experiment in Computer Science!  
Ada Lovelace spends days analyzing the collected data.  
Ada Lovelace analyzes data.

Jane Goodall has made significant breakthroughs in Primatology and published 6 papers.  
Personal information about Jane Goodall (age, nationality, specialty): 45, British, Primatology research  
Jane Goodall conducts a groundbreaking experiment in Primatology!  
Jane Goodall spends days analyzing the collected data.  
Jane Goodall analyzes data.

Elon Musk is a titan of the Technology industry, with 3 successful ventures reshaping global markets.  
Personal information about Elon Musk (age, nationality, specialty): 40, American, Technology industry  
Elon Musk successfully launches a startup in the Technology industry!  
Elon Musk negotiates lucrative deals for their company.  
Elon Musk networks.

Oprah Winfrey is a titan of the Media industry, with 5 successful ventures reshaping global markets.  
Personal information about Oprah Winfrey (age, nationality, specialty): 32, American, Media industry  
Oprah Winfrey successfully launches a startup in the Media industry!  
Oprah Winfrey negotiates lucrative deals for their company.  
Oprah Winfrey negotiates deals.

Jeff Bezos is a titan of the E-commerce industry, with 7 successful ventures reshaping global markets.  
Personal information about Jeff Bezos (age, nationality, specialty): 45, American, E-commerce industry  
Jeff Bezos successfully launches a startup in the E-commerce industry!  
Jeff Bezos negotiates lucrative deals for their company.  
Jeff Bezos negotiates deals.

Richard Branson is a titan of the Aviation industry, with 4 successful ventures reshaping global markets.  
Personal information about Richard Branson (age, nationality, specialty): 55, British, Aviation industry  
Richard Branson successfully launches a startup in the Aviation industry!  
Richard Branson negotiates lucrative deals for their company.  
Richard Branson negotiates deals.

Mark Zuckerberg is a titan of the Social Media industry, with 6 successful ventures reshaping global markets.  
Personal information about Mark Zuckerberg (age, nationality, specialty): 48, American, Social Media industry  
Mark Zuckerberg successfully launches a startup in the Social Media industry!  
Mark Zuckerberg negotiates lucrative deals for their company.  
Mark Zuckerberg launches a startup.

Average salary of these professionals is \$143500.00.

Professionals ranked by their salary:

1. Jeff Bezos - \$250000
2. Richard Branson - \$220000
3. Mark Zuckerberg - \$210000
4. Elon Musk - \$200000
5. Steve Jobs - \$180000
6. Oprah Winfrey - \$180000
7. Martha Stewart - \$150000
8. Albert Einstein - \$150000
9. Isaac Newton - \$140000
10. Leonhard Euler - \$130000
11. Stephen Hawking - \$130000
12. Georgia O'Keeffe - \$120000
13. Marie Curie - \$120000
14. Pablo Picasso - \$110000
15. Ada Lovelace - \$110000
16. Leonardo da Vinci - \$100000
17. Jane Goodall - \$100000
18. Frida Kahlo - \$95000
19. Vincent van Gogh - \$90000
20. Jackson Pollock - \$85000