



## Tema 1

# Sisteme Semantice

Draghici Andreea-Maria  
Inginerie Software  
IS 2.1

---

# Tema 1

---

## Contents

1. Descrierea rezolvării problemei .....	2
2. Rezultatele obținute .....	3
3. Concluzii .....	4
4. Cod sursă .....	4

# Tema 1

---

## 1. Descrierea rezolvării problemei

Proiectul constă în desenarea mai multor elemente construite într-un peisaj. Aceste elemente includ munți, soare, nori, case, copaci și un lac. Fiecare dintre aceste componente a fost desenată utilizând funcții individuale, fiecare funcție corespunzând unui element grafic.

- **Configurarea ferestrei de desenare:**
  - Am inițializat o fereastră de desen cu dimensiunile de 800x600 pixeli, având un fundal albastru deschis pentru a simula cerul.
  - Am setat viteza maximă de desenare pentru a eficientiza procesul.
- **Desenarea planului de iarbă:**
  - Utilizând funcția `draw_grass`, am desenat un dreptunghi verde la baza ferestrei, care simbolizează pajiștea pe care se vor construi restul elementelor.
- **Desenarea munților:**
  - Am desenat trei munți folosind funcția `draw_mountain`, care plasează triunghiuri mari, gri, cu un vârf alb ce sugerează zăpada. Acești munți sunt poziționați în fundal, oferind adâncime peisajului.
- **Desenarea soarelui și a razelor:**
  - Cu funcția `draw_sun`, am desenat un soare galben în centrul superior al ferestrei, compus dintr-un cerc și 12 raze. Acesta adaugă un element dinamic și vital peisajului.
- **Desenarea norilor:**
  - Norii au fost creați utilizând mai multe cercuri albe, suprapuse și combinate, folosind funcția `draw_cloud`. Acestea variază ca dimensiuni și sunt poziționați pe cer pentru a crea un efect de diversitate.
- **Desenarea caselor și copacilor:**
  - Am utilizat funcția `draw_scene` pentru a construi casele și copacii. Casele sunt compuse dintr-un pătrat pentru bază și un triunghi pentru acoperiș, în timp ce copacii sunt construiți dintr-un trunchi dreptunghiular și frunze sub formă de cerc. Fiecare casă este însoțită de un copac și a fost poziționată strategic la poalele munților.

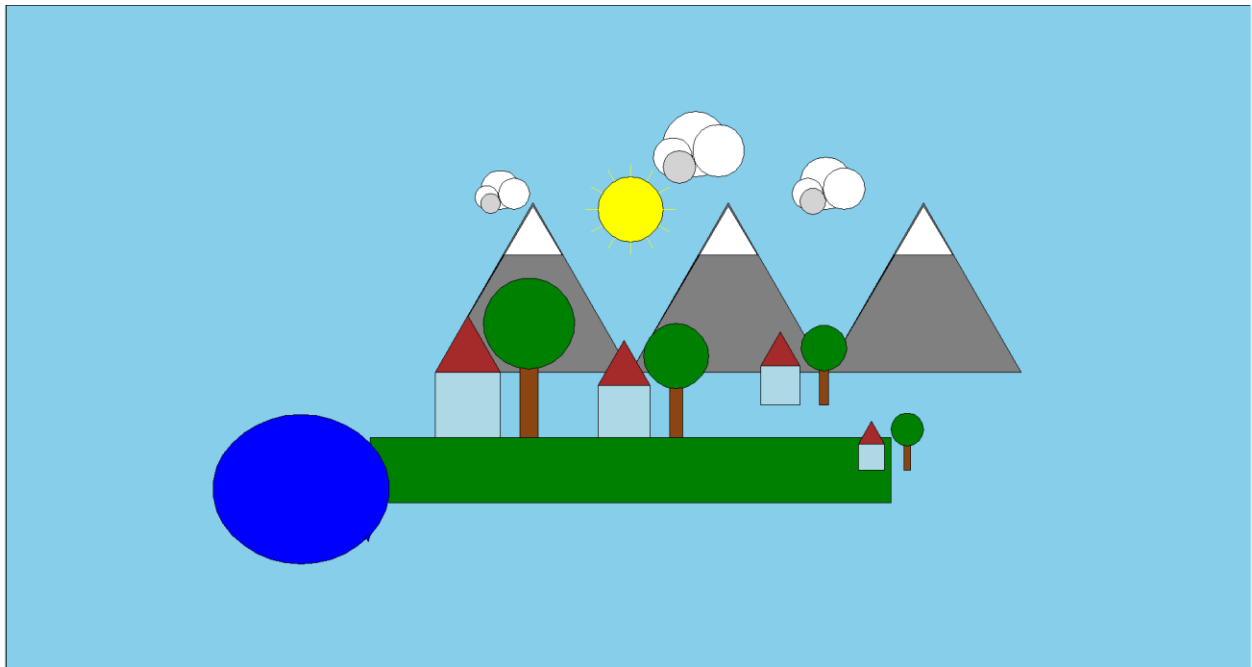
# Tema 1

---

- Desenarea lacului:
  - Lacul, desenat folosind funcția `draw_lake`, este plasat în partea stângă a peisajului, și are o formă ovală. Acesta adaugă un element de apă pentru a echilibra vizual scena.

## 2. Rezultatele obținute

- Obiectele (casele, munții, pomii și norii) sunt multiplicare pe același ecran, la poziții și dimensiuni diferite pentru a adăuga diversitate.
- Casele și copacii au fost desenate în mai multe instanțe, cu dimensiuni și poziții diferite.
- Fiecare casă este însoțită de un copac, iar aceste perechi sunt plasate strategic în scenă, fără a fi toate identice.
- Norii au fost desenați în mai multe instanțe, de dimensiuni diferite, și plasați pe cer.



# Tema 1

---

## 3. Concluzii

- Prin acest exercițiu, am reușit să aplic fundamentele de desenare grafică folosind librăria Turtle, astfel creând un peisaj. Soluția demonstrează cum forme simple precum cercurile, triunghiurile și pătratele pot fi combinate pentru a realiza o scenă landscape.
- Fiecare componentă a peisajului este desenată folosind funcții individuale, ceea ce face codul mai ușor de întreținut și reutilizat.

## 4. Cod sursă

*# Functia pentru desenarea unui patrat (folosit ca baza a casei)*

```
def draw_square(t, size):
```

```
    """
```

```
    Deseneaza un patrat cu dimensiunea specificata.
```

```
    Args:
```

```
        t (Turtle): Obiectul Turtle care deseneaza.
```

```
        size (int): Dimensiunea laturii patratului.
```

```
    """
```

```
    for _ in range(4): # Repetam de 4 ori pentru a desena un patrat
```

```
        t.forward(size) # Mergem inainte cu dimensiunea specificata
```

```
        t.left(90) # Rotim la stanga 90 de grade pentru a forma colturile
```

*# Functia pentru desenarea unui triunghi (folosit ca acoperisul casei)*

```
def draw_triangle(t, size):
```

```
    """
```

```
    Deseneaza un triunghi echilateral cu latura specificata.
```

```
    Args:
```

```
        t (Turtle): Obiectul Turtle care deseneaza.
```

```
        size (int): Dimensiunea laturii triunghiului.
```

```
    """
```

```
    for _ in range(3): # Repetam de 3 ori pentru a desena un triunghi
```

```
        t.forward(size) # Mergem inainte cu dimensiunea specificata
```

```
        t.left(120) # Rotim la stanga 120 de grade pentru a forma triunghiul echilateral
```

*# Functia pentru desenarea unei case (un patrat pentru baza + un triunghi pentru acoperis)*

```
def draw_house(t, size):
```

```
    """
```

```
    Deseneaza o casa compusa dintr-un patrat (baza) si un triunghi (acoperis).
```

```
    Args:
```

```
        t (Turtle): Obiectul Turtle care deseneaza.
```

```
        size (int): Dimensiunea casei (latura patratului).
```

```
    """
```

```
    t.fillcolor("lightblue") # Setam culoarea de umplere la albastru deschis pentru casa
```

```
    t.begin_fill() # Incepem umplerea formei patratului
```

```
    draw_square(t, size) # Desenam baza patrata a casei
```

# Tema 1

---

```
t.end_fill() # Terminam umplerea

# Mutam turtle la partea de sus a patratului pentru a incepe desenarea acoperisului
t.left(90)
t.forward(size)
t.right(90)

t.fillcolor("brown") # Setam culoarea de umplere la maro pentru acoperis
t.begin_fill() # Incepem umplerea formei triunghiului
draw_triangle(t, size) # Desenam acoperisul triunghiular
t.end_fill() # Terminam umplerea

# Functia pentru desenarea unui copac complet (trunchi + frunze)
def draw_tree(t, size):
    """
    Deseneaza un copac compus dintr-un trunchi dreptunghiular si un cerc (frunzele).

    Args:
        t (Turtle): Obiectul Turtle care deseneaza.
        size (int): Dimensiunea copacului.
    """
    # Desenam trunchiul copacului (un dreptunghi maro)
    t.fillcolor("#8B4513") # Setam culoarea de umplere la maro pentru trunchi
    t.begin_fill() # Incepem umplerea formei dreptunghiului
    for _ in range(2): # Repetam de 2 ori pentru a desena un dreptunghi
        t.forward(size * 0.2) # Partea scurta a dreptunghiului (latimea trunchiului)
        t.left(90) # Rotim la stanga 90 de grade
        t.forward(size) # Partea lunga a dreptunghiului (inaltimea trunchiului)
        t.left(90) # Rotim din nou la stanga
    t.end_fill() # Terminam umplerea

    # Mutam turtle in pozitia de unde vor fi desenate frunzele
    t.penup() # Ridicam stiloul pentru a nu desena in timp ce ne mutam
    t.left(90) # Rotim turtle in sus
    t.forward(size) # Mutam turtle deasupra trunchiului
    t.right(90) # Rotim turtle la dreapta pentru a pregati desenarea frunzelor

    # Mutam turtle putin mai sus si inainte pentru a centra frunzele
    t.forward(size * 0.1) # Mutam turtle usor inainte
    t.right(90) # Rotim turtle la dreapta
    t.forward(size * 0.25) # Mutam turtle in sus pentru a centra frunzele
    t.left(90) # Rotim turtle inapoi la pozitia normala

    # Desenam frunzele copacului (un cerc verde)
    t.pendown() # Coboram stiloul pentru a incepe desenarea
    t.fillcolor("green") # Setam culoarea de umplere la verde pentru frunze
    t.begin_fill() # Incepem umplerea cercului
    t.circle(size * 0.5) # Desenam cercul care reprezinta frunzele
    t.end_fill() # Terminam umplerea

# Functia pentru desenarea soarelui cu raze
def draw_sun(t, x, y, size):
    """
```

# Tema 1

Deseneaza un soare la pozitia specificata, format dintr-un cerc si raze.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

x (int): Coordonata x a pozitiei soarelui.

y (int): Coordonata y a pozitiei soarelui.

size (int): Raza soarelui (cercul).

"""

*# Mutam turtle la pozitia specificata pentru soare*

*t.penup() # Ridicam stiloul pentru a nu desena in timpul mutarii*

*t.goto(x, y) # Mutam turtle la coordonatele x si y specificate*

*t.pendown() # Coboram stiloul pentru a incepe desenarea soarelui*

*# Desenam cercul pentru soare*

*t.fillcolor("yellow") # Setam culoarea de umplere la galben pentru soare*

*t.begin\_fill() # Incepem umplerea cercului*

*t.circle(size) # Desenam cercul care reprezinta soarele*

*t.end\_fill() # Terminam umplerea*

*# Desenam razele soarelui*

*t.pencolor("yellow") # Setam culoarea stiloului la galben pentru raze*

*t.penup() # Ridicam stiloul din nou pentru a ne pozitiona pentru raze*

*for \_ in range(12): # Desenam 12 raze in jurul soarelui*

*t.goto(x, y + size) # Mutam turtle la partea de sus a soarelui*

*t.pendown() # Coboram stiloul pentru a desena raza*

*t.forward(size + 20) # Desenam raza in afara cercului*

*t.penup() # Ridicam stiloul dupa ce am desenat raza*

*t.goto(x, y) # Revenim la centrul soarelui*

*t.right(30) # Rotim turtle cu 30 de grade pentru urmatoarea raza*

*# Resetam culoarea stiloului la negru pentru desene viitoare*

*t.pencolor("black")*

*# Functia pentru desenarea unui munte cu varf de zapada*

**def** draw\_mountain(t, x, y, size):

"""

Deseneaza un munte sub forma de triunghi la coordonatele specificate.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

x (int): Coordonata x a bazei muntelui.

y (int): Coordonata y a bazei muntelui.

size (int): Lungimea laturii muntelui (triunghiului).

"""

t.penup()

t.goto(x, y)

t.pendown()

*# Desenam muntele principal ca un triunghi gri*

*t.fillcolor("grey") # Setam culoarea de umplere la gri pentru munte*

*t.begin\_fill() # Incepem umplerea triunghiului*

*for \_ in range(3): # Desenam cele 3 laturi ale triunghiului*

*t.forward(size)*

# Tema 1

---

```
t.left(120)
t.end_fill() # Terminam umplerea
```

*# Desenam varful cu zapada al muntelui*

```
t.fillcolor("white")
t.begin_fill()
t.goto(x + size * 0.35, y + size * 0.6) # Desenam partea stanga a zapezii
t.goto(x + size * 0.65, y + size * 0.6) # Desenam partea dreapta a zapezii
t.goto(x + size * 0.5, y + size * 0.85) # Desenam varful zapezii
t.goto(x, y) # Revenim la baza muntelui
t.end_fill() # Terminam umplerea
```

*# Functia pentru desenarea unui lac (forma ovala cu reflexii)*

```
def draw_lake(t, x, y, width, height):
```

```
    """
```

Deseneaza un lac in forma de oval la coordonatele specificate.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

x (int): Coordonata x de pornire.

y (int): Coordonata y de pornire.

width (int): Latimea lacului.

height (int): Inaltimea lacului.

```
    """
```

```
t.penup()
t.goto(x, y)
t.pendown()
t.fillcolor("blue")
t.begin_fill()
```

*# Desenam un oval combinand doua arcuri*

```
t.setheading(45)
for _ in range(2):
    t.circle(width, 90)
    t.circle(height, 90)
t.end_fill()
```

*# Functia pentru desenarea unui nor cu umbra*

```
def draw_cloud(t, x, y, size):
```

```
    """
```

Deseneaza un nor la coordonatele specificate, compus din mai multe cercuri.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

x (int): Coordonata x de pornire.

y (int): Coordonata y de pornire.

size (int): Dimensiunea norului (raza cercului de baza).

```
    """
```

```
t.penup()
t.goto(x, y)
t.pendown()
```

*# Desenam cercurile care formeaza norul*



# Tema 1

---

```
t.fillcolor("white")
t.begin_fill()
t.circle(size)
t.end_fill()
```

```
t.penup()
t.goto(x + size * 0.7, y)
t.pendown()
t.begin_fill()
t.circle(size * 0.8)
t.end_fill()
```

```
t.penup()
t.goto(x - size * 0.7, y)
t.pendown()
t.begin_fill()
t.circle(size * 0.6)
t.end_fill()
```

*# Desenam umbra norului*

```
t.penup()
t.goto(x - size * 0.5, y - size * 0.2)
t.pendown()
t.fillcolor("lightgrey")
t.begin_fill()
t.circle(size * 0.5)
t.end_fill()
```

*# Functia pentru desenaarea unui plan de iarba (un dreptunghi verde)*

**def** draw\_grass(t, start\_x, start\_y, width, height):

"""

Deseneaza un plan de iarba sub forma de dreptunghi verde.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

start\_x (int): Coordonata x de pornire.

start\_y (int): Coordonata y de pornire.

width (int): Latimea planului de iarba.

height (int): Inaltimea planului de iarba.

"""

```
t.penup()
t.goto(start_x, start_y)
t.pendown()
```

```
t.fillcolor("green")
t.begin_fill()
```

**for** \_ in range(2): *# Desenam un dreptunghi repetand de 2 ori*

t.forward(width)

t.left(90)

t.forward(height)

t.left(90)

```
t.end_fill()
```

# Tema 1

---

*# Functia pentru desenearea unei scene complete (o casa si un copac)*

```
def draw_scene(t, x, y, house_size, tree_size):
```

```
    """
```

Deseneaza o scena completa, compusa dintr-o casa si un copac.

Args:

t (Turtle): Obiectul Turtle care deseneaza.

x (int): Coordonata x pentru pozitionarea scenei.

y (int): Coordonata y pentru pozitionarea scenei.

house\_size (int): Dimensiunea casei (latura).

tree\_size (int): Dimensiunea copacului.

```
    """
```

```
t.penup()
```

```
t.goto(x, y)
```

```
t.pendown()
```

```
draw_house(t, house_size) # Desenam casa la pozitia specificata
```

*# Pozitionam copacul la dreapta casei*

```
t.penup()
```

```
t.goto(x + house_size + 30, y) # Mutam la dreapta casei
```

```
t.pendown()
```

```
draw_tree(t, tree_size) # Desenam copacul
```

```
import turtle
```

```
def run_setup() :
```

```
    """
```

Functia principala care initializeaza ecranul si deseneaza peisajul complet cu munti, soare, case, copaci si un lac.

```
    """
```

*# Setup pentru Turtle*

```
screen = turtle.Screen() # Cream o noua fereastră pentru desenare
```

```
screen.setup(width=800, height=600) # Setam dimensiunea ferestrei la 800x600 pixeli
```

```
screen.bgcolor("skyblue") # Setam culoarea de fundal a ferestrei la albastru deschis (cer)
```

```
t = turtle.Turtle() # Cream un nou obiect Turtle pentru desenare
```

```
t.speed(0) # Setam viteza de desenare la maxim (cea mai rapida)
```

*# Desenam un plan de iarba (un dreptunghi verde) la baza ecranului*

```
draw_grass(t, -400, -250, 800, 100) # Iarba va acoperi partea de jos a ecranului (un dreptunghi verde)
```

*# Desenam muntii in fundal*

```
draw_mountain(t, -300, -50, 300) # Primul munte (mare) in stanga
```

```
draw_mountain(t, 0, -50, 300) # Al doilea munte (central)
```

```
draw_mountain(t, 300, -50, 300) # Al treilea munte (dreapta)
```

*# Desenam soarele*

```
draw_sun(t, 0, 150, 50) # Soare in centru, sus, de dimensiune medie
```

*# Desenam trei nori de dimensiuni diferite in diverse pozitii*

```
draw_cloud(t, -200, 200, 30) # Primul nor (mic) in stanga
```

```
draw_cloud(t, 100, 250, 50) # Al doilea nor (mare) in centru
```

```
draw_cloud(t, 300, 200, 40) # Al treilea nor (mediu) in dreapta
```

# Tema 1

---

```
# Desenam cate o scena completa cu o casa si un copac in diverse pozitii
draw_scene(t, -300, -150, 100, 140) # Prima casa (mare) si copac, in stanga
draw_scene(t, -50, -150, 80, 100) # A doua casa (medie) si copac, in centru
draw_scene(t, 200, -100, 60, 70) # A treia casa (mica) si copac, in dreapta
draw_scene(t, 350, -200, 40, 50) # A patra casa (foarte mica) si copac, mai jos in dreapta

# Desenam un lac in partea stanga a ecranului
draw_lake(t, -400, -300, 100, 150) # Lacul va fi oval si va fi in partea de jos stanga

# Finalizam desenul
turtle.done() # Inchidem fereastra Turtle dupa ce desenul este complet

if __name__ == '__main__':
    run_setup() # Apelam functia principala
```