

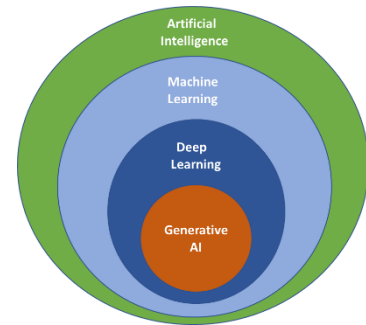
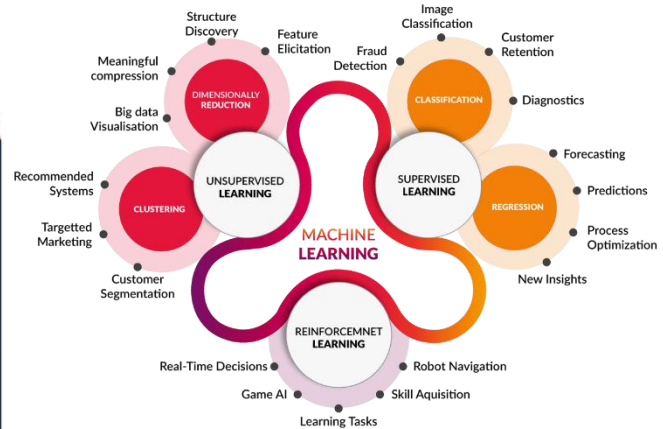
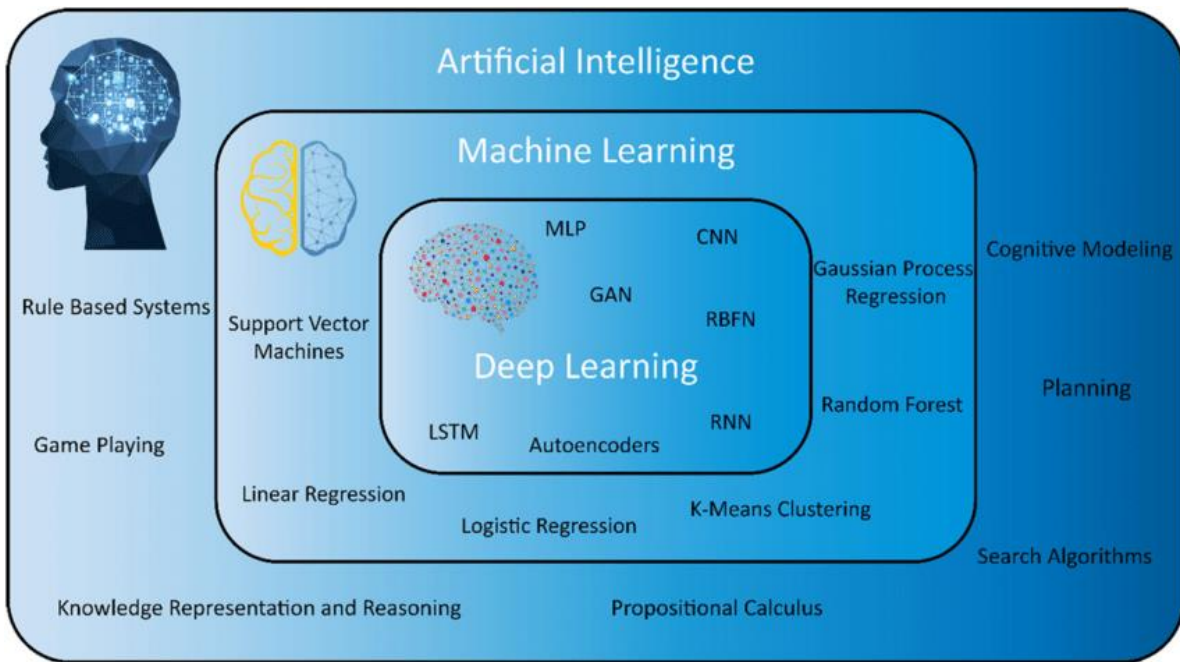


ChatGPT az építőmérnöki tanteremben

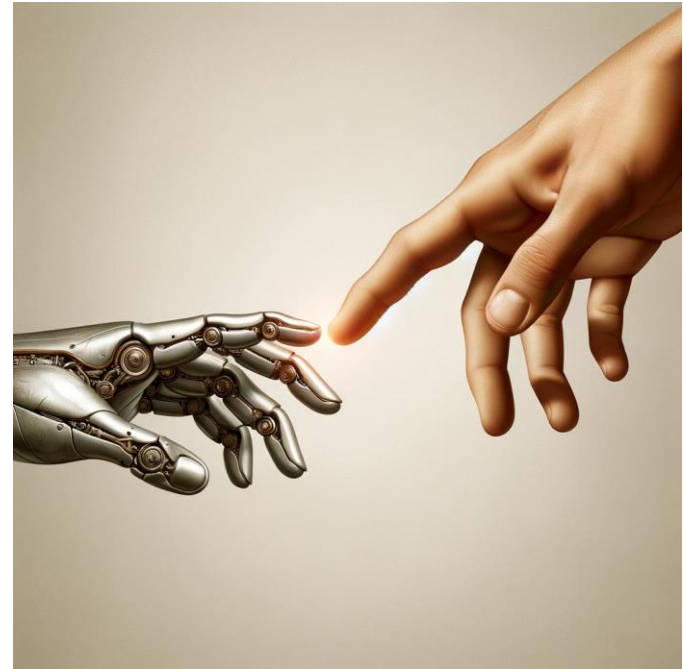
Barsi Árpád



AI, ML, NN, DL, ...

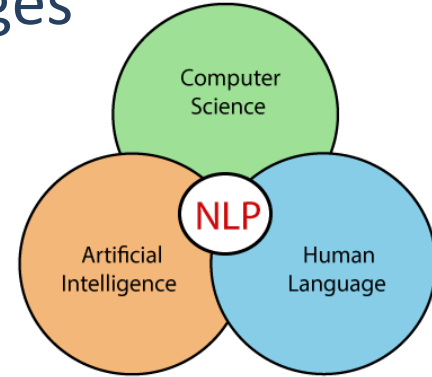


- NLP és ChatGPT
- Hogyan használju(n)k ChatGPT-t?
 - Prompt engineering
 - Pair programming



NLP és ChatGPT

- **NLP** (Természetes nyelvfeldolgozás): a mesterséges intelligencia egy ága, amely arra összpontosít, hogy a számítógépek képesek legyenek értelmezni, feldolgozni és generálni az emberi nyelvet.
- **ChatGPT**: a ChatGPT egy előre betanított nyelvi modell, amit az OpenAI fejlesztett ki, és képes természetes párbeszédre, szöveges válaszok generálására és a nyelvfeldolgozási feladatok elvégzésére.





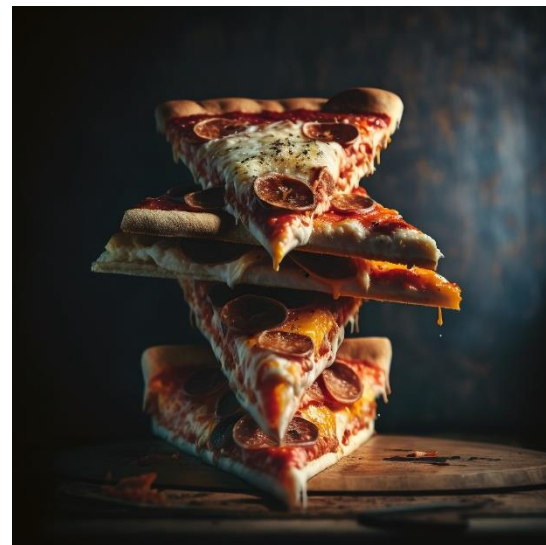
Mi a „prompt engineering”?

A **prompt** az az input vagy utasítás, amit a nyelvi modellbe táplálunk annak érdekében, hogy a legjobb választ kapjuk. Minél specifikusabb, árnyaltabb a promptunk, annál jobb megoldást (választ) fogunk kapni a rendszertől. A válasz lehet szöveg, kép, hang, videó vagy akár egy kódsor is.



stack of pizza slices

*stack of delicious pizza
slices, food photo style,
close up, photo, 8k,
rustic, cinematic shot,
dramatic lighting*





Saját próbálkozás

- generate an image of a stack of delicious pizza slices, food photo style, close up, photo, 8k, rustic, cinematic shot, dramatic lighting



Hatékony promptok

- **Érthetőség** (Clarity)
 - Ahelyett hogy „Beszélj a kutyákról”, inkább "Adj részletes leírást a házikutyák jellemzőiről, viselkedéséről és gondozásáról."
- **Kontextus** (Context)
 - "Fordítsd le a következő angol szöveget franciára: *,Hello, how are you?'*"
- **Pontosság** (Precision)
 - „Sorold fel a világ 10 legnépesebb országát!"
- **Szerep megjelölése** (Role-play)
 - "Történészként magyarázd el az amerikai polgárháború jelentőségét."



Prompt „debugging”

- A hangszín és a formalitás beállítása
 - "Magyarázd el nekem, mintha öt éves lennék"
- A részletek finomítása
 - Olyan szavak, mint „röviden” vagy „részletesen” hozzáadása a kérdéshez
- Kísérletezés és iteráció



Prompt példa 1

BA **You**

A lehető legvilágosabb és specifikus utasításokkal kell kifejeznie, hogy mit szeretne egy modelltől. Ez elvezeti a modellt a kívánt kimenet felé, és csökkenti annak esélyét, hogy irreleváns vagy helytelen válaszokat kapjon. Ne keverje össze a világos felszólítás írását a rövid felszólítással. Sok esetben a hosszabb promptok egyértelműbbé és kontextusosabbá teszik a modellt, ami részletesebb és relevánsabb kimenetekhez vezethet.

Foglald össze egyetlen mondatban a fenti szöveget!

 **ChatGPT**

A hatékony modellutasítások világosak, specifikusak és részletesek, ami segít elkerülni az irreleváns vagy helytelen válaszokat, és nem feltétlenül rövidek.





Prompt példa 2

BA

You

Hozz létre egy listát három kitalált könyvcímből a szerzőkkel és műfajukkal együtt. Add meg őket JSON formátumban a következő kulcsokkal: könyv_azonosítója, cím, szerző, műfaj

```
[
  {
    "könyv_azonosítója": "K001",
    "cím": "Az Elfeledett Város",
    "szerző": "Anna Kovács",
    "műfaj": "Fantasy"
  },
  {
    "könyv_azonosítója": "K002",
    "cím": "Az Idő Kapui",
    "szerző": "Péter Nagy",
    "műfaj": "Sci-fi"
  },
  {
    "könyv_azonosítója": "K003",
    "cím": "Rejtélyes Emlékek",
    "szerző": "Erika Varga",
    "műfaj": "Krimi"
  }
]
```

Prompt példa 3

BA You

Egy csésze tea elkészítése egyszerű! Először is fel kell forralnia egy kis vizet. Amíg ez megtörténik, fogjon egy csészét, és tegyen bele egy teászacskót. Ha a víz kellően felforrósodott, csak öntse rá a teászacskót. Hagyja állni egy kicsit, hogy a tea beázhasson. Néhány perc múlva vegye ki a teászacskót. Ízlés szerint cukrot vagy tejet adhatunk hozzá. És ez az! Van egy csésze finom teája, amelyet élvezhet.

Ha utasítássorozatot tartalmaz,
írd át ezeket az utasításokat a következő formátumban:

1. lépés - ...

2. lépés - ...

...

N. lépés - ...

Ha a szöveg nem tartalmaz utasítássorozatot,
egyszerűen írd be: "Nincs lépés megadva".



ChatGPT

1. lépés - Forraljon fel egy kis vizet.
2. lépés - Vegyen egy csészét, és tegyen bele egy teászacskót.
3. lépés - Amikor a víz felforr, öntse rá a teászacskóra.
4. lépés - Hagyja állni a teát néhány percig, hogy beázhasson.
5. lépés - Vegye ki a teászacskót a csészéből.
6. lépés - Ízlés szerint adja hozzá a cukrot vagy a tejet.
7. lépés - Élvezze a finom teáját.



Prompt példa 4

BA

You

Válaszolj következetes stílusban!

<gyerek>: Taníts meg a türelemre!

<nagyszülő>: A legmélyebb völgyet vajú folyó szerény forrásból fakad; a legnagyobb szimfónia egyetlen hangból ered; a legbonyolultabb faliszőnyeg magányos fonállal kezdődik.

<gyerek>: Taníts a rugalmasságról!



ChatGPT

<nagyszülő>: A bölcs fű a szélben hajlik, de sosem törik meg; a patak kanyarog az akadályok között, de mindig eléri a tengerét; az ügyes művész hibáit mesteri alkotássá formálja, így születik meg a tökéletes mű.

Prompt példa 5

BA

You

Döntsd el, hogy a diák megoldása helyes-e vagy sem.

Kérdés:

Napelem telepítést tervezek és segítségre van szükségem az anyagiak kidolgozásához.

- A telek ára 100 dollár/négyzetméter
- Napelemeket 250 dollár/négyzetméter áron tudok venni
- A karbantartási szerződés szerint évi 100 ezer dollárba kerül a lakásra, és további 10 dollár négyzetméterenként.

Mennyi a teljes költség az első működési évben a négyzetméter függvényében.

A diák megoldása:

Legyen x a berendezés mérete négyzetméterben.

Költségek:

1. Telekköltség: $100x$
 2. Napelem költség: $250x$
 3. Fenntartási költség: $100\ 000 + 100x$
- Teljes költség: $100x + 250x + 100\ 000 + 100x = 450x + 100\ 000$



ChatGPT

A diák megoldása helytelen. A fenntartási költségek kiszámítása hibás, mert a diák a $100x$ összeget használta, ami nem felel meg a "10 dollár négyzetméterenként" feltételnek. A helyes fenntartási költség kiszámítása $10x$ dollár lenne négyzetméterenként, nem $100x$.

A helyes összköltség kiszámítása így néz ki:

1. Telekköltség: $100x$ dollár
2. Napelem költség: $250x$ dollár
3. Fenntartási költség: $100\ 000$ dollár + $10x$ dollár

Teljes költség: $100x + 250x + 100\ 000 + 10x = 360x + 100\ 000$.



BA

You

A tárgy célja alapvető áttekintést adni a CAD rendszerekről, és azok alkalmazási területeiről. A hallgatók a gyakorlatokon elsajátíthatják azokat a síkban alkalmazt parancsokat, amelyek lehetővé teszik alapvető tervezési feladatok végrehajtását kurzus során külön tárgyalásra kerül a réteghasználat, a blokk készítés, a feliratozás valamint a méretezés. A nyomtatási beállítások ismertetése szintén segítik a későbbi tervezési feladatok megvalósítását. A tárgy célja, hogy a hallgatók átlássák a CAD rendszerek képességeit, lehetőségeit és az általuk kiszolgált alkalmazásokat. Továbbá a gyakorlatok során az alapvető 3D (térbeli) szerkesztési módszerek is tárgyalásra kerülnek, ezzel megalapozva későbbi 3D tervezést igénylő tárgyakat, BIM rendszer szerkesztési feladatait.

A fenti szöveg a BME Építőmérnöki Karon oktatott Építőmérnöki CAD tantárgy rövid leírása. Magyarázd el egy vidéki idős parasztbácsinak, hogy mit tanul az unokája, amikor ezzel a tantárggyal foglalkozik Budapesten, mintha az unokája beszélné.



ChatGPT

Kedves Nagypapa!

Budapesten egy nagyon érdekes tantárgyat tanulok, amit CAD-nek hívnak. Ez egy olyan számítógépes program, ami segít nekünk terveket készíteni. Pontosan olyan, mint amikor te rajzolsz a terveket a gazdasághoz, csak én ezt a számítógépen csinálom.

Megtanulom, hogyan lehet a számítógépen különböző rajzokat készíteni. Például megtanulom, hogyan rajzoljak egy házat, hídat vagy bármilyen építményt. Ebben a programban lehetőségem van mindent pontosan méretre rajzolni, és még azt is beállíthatom, milyen anyagokból épüljön az építmény.

Tanulunk a rajzokat rétegekre osztani is, ami olyan, mint amikor több papírra rajzolsz, és aztán egymásra teszed őket. Így könnyebb megérteni a terv minden részletét. Tanulunk arról is, hogyan írunk fel fontos dolgokat a tervekre, mint például a méretek vagy a nevek.

A legizgalmasabb része, hogy 3D-ben is tanulunk tervezni. Ez azt jelenti, hogy a számítógépen olyan, mintha tényleg létezne az, amit rajzolunk, és körbe lehet nézni. Ez nagyon hasznos, mert így pontosan látjuk, hogyan fog kinézni valójában az épület.

Ez a tanulás nagyon fontos, mert így megtanulhatom, hogyan tervezek biztonságosan és hatékonyan. Remélem, egyszer megmutathatom neked is, hogyan működik ez a program, biztos tetszene!

Szeretettel,

Unokád

Prompt példa 7

BA

You

"Kelt mint itt Szanfranciskóban uralkodásunk elr
Igen tisztelt kiráj úr, kedves mamája és T. neje őfel
vettem és kibontám.

Ezennel felelek tisztelettel értesíteni!

Szíves mekhívásá
gondtalanít, van s
tétlenség. Én váró
formalinság. De v
kicsiny, mégiscsa
vissza uralkodás
sem lepne meg, h

találkoznék sétálva. Mer a nép bezárt engem a szí
zacskót ragasztani, amint egyéb bezárás esetén e

Az eredeti szöveg helyesírási hibáit javítsd, a stílus
Ezután alkoss a szövegből Shakespeare stílusában



ChatGPT

A szöveg helyesírási hibáinak javítása után, a stílus megőrzésével:

"Kelt, mint itt San Franciscóban uralkodásunk elmúltával, magamkezőleg.

Igen tisztelt király úr, kedves mamája és T. neje Őfelsége! Mai nappal kelt, fent levelét
vettem és kibontám.

San Franciscóban uralkodtam, elmúlt az időm,
Majd király meghív, de a városi élet hív szívem.
Emlékszem uralkodásom ragyogó napjaira,
De szabadság vágya vezet, nem a királyi udvar falai.

napsugaras idejére, amely felvirágoztatta az országot... És az sem lepne meg, ha
Almirában megőrkítenék fénykoromat és a főtéren szobrommal találkoznék sétálva.
Mert a nép bezárt engem a szívébe, és jó, hogy mégsem kell zacskót ragasztani, amint
egyéb bezárás esetén ez szokás."

Pair programming 1

- Show me how to iterate across a list in Matlab

 ChatGPT

In MATLAB, iterating across a list (or more commonly, an array) is typically done using a `for` loop. Here's an example of how you can iterate over the elements of a list:


Suppose you have

```
matlab
```

```
myList = [
```

You can iterate over this list using a `for` loop as follows:

```
matlab
```

 Copy code

```
for i = 1:length(myList)
    currentValue = myList(i);
    % Do something with currentValue
    disp(currentValue)
end
```

In this example, the `for` loop starts with `i = 1` and increments `i` by 1 on each iteration, continuing until `i` reaches the length of `myList`. On each iteration, `currentValue` is set to the element of `myList` at the current index `i`, and then you can perform any desired operation with this `currentValue`. In this case, it simply displays the value using `disp()`.

Pair programming 2

- write code to iterate across a list in Python

python

Copy code

```
my_list = [1, 2, 3, 4, 5]
```

```
for item in my_list:  
    print(item)
```



```
my_list = [1, 2, 3, 4, 5]
```

```
for item in my_list:  
    print(item)
```

Pair programming 3

- Show me variations how to iterate across a list in Python

1. Basic Iteration:

python

```
my_list = [1, 2, 3, 4, 5]
for item in my_list:
    print(item)
```

2. Iterating with Index:

python

```
my_list = [1, 2, 3, 4, 5]
for index, item in enumerate(my_list):
    print(f"Index: {index}, Value: {item}")
```

3. List Comprehension:

python

```
my_list = [1, 2, 3, 4, 5]
new_list = [item * 2 for item in my_list]
print(new_list)
```

4. Using `map` Function:

python

```
my_list = [1, 2, 3, 4, 5]
def square(x):
    return x * x
squared_list = map(square, my_list)
print(list(squared_list))
```

5. Using `filter` Function (for filtering the list based on a condition):

python

Copy code

7. Using a While Loop:

python

Copy code

```
my_list = [1, 2, 3, 4, 5]
index = 0
while index < len(my_list):
    print(my_list[index])
    index += 1
```



```
for item in reversed(my_list):
    print(item)
```

Pair programming 4

python Copy code

```
# Node class for a doubly linked list
class Node:
    def __init__(self, data):
        self.data = data # Store data
        self.next = None # Initialize next as null
        self.prev = None # Initialize prev as null

# Doubly Linked List class
class DoublyLinkedList:
    def __init__(self):
        self.head = None # Initialize head of list as null

# Function to add a node at the front of the list
def push(self, new_data):
    new_node = Node(new_data) # Create a new node
    new_node.next = self.head # Make next of new node as head
    if self.head is not None:
        self.head.prev = new_node # Change previous of head node to
    self.head = new_node # Move the head to point to the new node

# Function to insert a new node after a given node
def insertAfter(self, prev_node, new_data):
    if prev_node is None:
        print("This node doesn't exist in DLL")
        return
    new_node = Node(new_data) # Create a new node
    new_node.next = prev_node.next # Make next of new node as next
    prev_node.next = new_node # Make next of prev_node as new_node
    new_node.prev = prev_node # Make previous of new_node as prev
    if new_node.next is not None:
        new_node.next.prev = new_node # Change previous of new_node

# Function to add a node at the end of the list
def append(self, new_data):
    new_node = Node(new_data) # Create a new node
    new_node.next = None # This new node is going to be the last node
    if self.head is None: # If the Linked List is empty, then make
        new_node.prev = None
        self.head = new_node
        return
    last = self.head
    while last.next: # Traverse till the last node
        last = last.next
    last.next = new_node # Change the next of last node
    new_node.prev = last # Make last node as previous of new node

# Function to print the doubly linked list from the head
def printlist(self):
    temp = self.head
    while temp:
        print(temp.data, end=" ")
        temp = temp.next
```



You

You are an expert at writing clear, concise, Python code.

priming

Create a doubly linked list.

question

Insert comments for each line of code.

decorator

Pair programming 5

• Kód

Változtatások:

1. A függvény deklarációjában hiányzik a kettőspont (`:`) a függvény neve után. Ezt hozzáadtam.
2. Az eredeti kódban a `for` ciklus a `range()` függvényt használja a lista hosszának meghatározására, és indexeket használ a lista elemeinek elérésére. Ez a megközelítés működik, de nem a legidiomatikusabb módja a Python listák bejárásának.
3. A javított verzióban közvetlenül iterálok az `array` elemein, ami egyszerűbb és tisztább kódhoz vezet. Ebben az esetben az `item` változó minden iterációban a lista következő elemét kapja meg. Ezáltal elkerüljük az indexelés szükségességét, és a kód könnyebben olvasható és karbantartható lesz.

Ez a változtatás azért jobb, mert a Python listák közvetlen iterálása egyszerűbb, hatékonyabb és jobban kifejezi a kód szándékát.

BA

You
Szerintem ne

```
def func_x(arr):  
    for i in range(len(arr)):  
        print(array[i])
```

Magyarázd el

ákon való

éppen néz ki:

Copy code



Pair programming 6

• Kód egyszerűsítése

BA

You


Tudod egyszerűsíteni ezt a kódot?

```
class Node:
    def __init__(self, dataval=None):
        self.dataval = dataval
        self.nextval = None
```

```
class SLinkedList:
    def __init__(self):
        self.headval = None
```

```
list1 = SLinkedList()
list1.headval = Node("Mon")
e2 = Node("Tue")
e3 = Node("Wed")
list1.headval.nextval = e2
e2.nextval = e3
```

Magyarázd el részletesen, hogy mit és miért változtattál!

python  Copy code

```
class Node:
    def __init__(self, dataval=None):
        self.dataval = dataval
        self.nextval = None
```

```
class SLinkedList:
    def __init__(self):
        self.headval = None
```

```
    def insert(self, newdata):
        newNode = Node(newdata)
        if self.headval is None:
            self.headval = newNode
            return
        last = self.headval
        while last.nextval:
            last = last.nextval
        last.nextval = newNode
```

```
# Létrehozok egy új SLinkedList példányt
list1 = SLinkedList()
```

```
# Hozzáadok elemeket a listához az insert függvény használatával
list1.insert("Mon")
list1.insert("Tue")
list1.insert("Wed")
```

 ChatGPT

A kód, amit megadtál, egy egyszerű láncolt lista (single linked list) implementációt mutat be Pythonban. A kód már elég egyszerű, de az inicializálás és az elemek hozzáadása egy kicsit tisztábbá tehető a következő változtatásokkal:

1. Hozzáadok egy `insert` függvényt az `SLinkedList` osztályhoz, amely lehetővé teszi az elemek egyszerűbb hozzáadását a listához.
2. Az új elemek hozzáadását a `SLinkedList` osztály `insert` függvényével végzem, ami javítja az olvashatóságot és csökkenti a redundanciát.

Pair programming 7

- Hibák

BA

You

```
%% faktoriális kl  
val = 0;  
n = 5;  
for i = 1:n  
    val = val * i;  
end  
disp([num2str(n)  
  
Magyarázd el ho
```

A helyes megoldás az, hogy inicializáljuk `val`-t 1-re, mivel a faktoriális számításnál az 1 a szorzás kezdőértéke (azaz $n! = n * (n-1) * \dots * 2 * 1$). Így a javított kód:

matlab

Copy code

```
%% faktoriális klasszikusan  
val = 1; % helyes kezdeti érték  
n = 5;  
for i = 1 : n  
    val = val * i;  
end  
disp([num2str(n) '!' = ' num2str(val)])
```



Ebben a javított verzióban `val` kezdeti értéke 1, így a ciklus helyesen számolja ki az `n` faktoriálisát.

ke 5. Azonban
t. A hiba a

Copy code

ális
nulla
ációk

- Unit tes

BA

You

Írj unit test-eket a köv

```
function val = faktoria
```

```
val = 1;
```

```
for i = 1:n
```

```
    val = val * i;
```

```
end
```

Készíts futtatható scri

Tegy megjegyzéseket

1. Tesztosztály létrehozása

Először írjuk meg a tesztosztályt ``faktorialisunkTest.m`` néven:

matlab

Copy code

```
classdef faktorialisunkTest < matlab.unittest.TestCase
```

```
    % Tesztosztály a faktorialisunk függvényhez
```

```
    methods (Test)
```

```
        function
```

```
            % Te
```

```
            actu
```

```
            expe
```

```
            test
```

```
        end
```

```
        function
```

```
            % Te
```

```
            actu
```

```
            expe
```

```
            test
```

```
        end
```

```
        function
```

```
            % Te
```

```
            actu
```

```
            expe
```

```
            test
```

```
        end
```

```
    end
```

```
end
```

2. Futtatható script létrehozása a tesztekhez

Most hozzunk létre egy scriptet ``runFaktorialisunkTests.m`` néven, amely futtatja a tesztek:

matlab

Copy code

```
% Script a faktorialisunk tesztosztályának futtatására
```

```
% Létrehoz egy teszteset példányt
```

```
tests = matlab.unittest.TestSuite.fromClass(?faktorialisunkTest);
```

```
% Futtatja a tesztek
```

```
results = run(tests);
```

```
% Megjeleníti a teszteredményeket
```

```
disp(table(results))
```



• Unit test az `assert` függvény

Ez a script három különböző tesztesetet tartalmaz:

1. **0 Faktoriális Tesztelése:** Ellenőrzi, hogy a 0 faktoriálisa 1-e.
2. **Pozitív Egész Szám Faktoriálisának Tesztelése:** Ellenőrzi, hogy a 5 faktoriálisa 120-e.
3. **1 Faktoriális Tesztelése:** Ellenőrzi, hogy az 1 faktoriálisa 1-e.

Ha bármelyik `assert` állítás hamis, akkor a script hibaüzenetet generál és leáll. Ha minden állítás igaz, akkor a script "Minden teszt sikeresen lefutott." üzenetet jelenít meg.

Itt van egy példa scriptre, amely teszteli a `faktorialisunk` függvényt:

matlab

Copy code

```
% Teszteli az 1 faktoriálisa
expectedValue = 1;
actualValue = faktorialisunk(1);
assert(actualValue == expectedValue, '1 faktoriálisa 1-nek kell lennie.

disp('Minden teszt sikeresen lefutott.');
```



Készíts futtatható script-et a tesztesetek vizsgálat
Tégy megjegyzéseket a teszt-esetekhez, hogy láth

BA

You

Írd át az alábbi Python-kódot Matlab-ra!

```
# Returns index of x in arr if present, else -1
def binary_search(arr, low, high, x):
    # Check base case
    if high >= low:
        mid = (high + low) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binary_search(arr, low, mid - 1, x)
        else:
            return binary_search(arr, mid + 1, high, x)
    else:
        return -1

# Test array
arr = [ 2, 3, 4, 10, 40 ]
x = 10

# Function call
result = binary_search(arr, 0, len(arr)-1, x)

if result != -1:
    print("Element is present at index", str(result))
else:
    print("Element is not present in array")
```

k nyelvre

A MATLAB kódban figyelni kell a következőkre:

1. MATLAB indexelése 1-től kezdődik, ellentétben a Pythonnal.
2. A `floor` függvényt használtam a középérték meghatározásához, mert a MATLAB-ban nincs egész osztás operátor.
3. A `print` helyett a `fprintf` függvényt használtam.
4. Az `if` és `else` blokkokban a MATLAB kódban a `end` szót kell használni.

Pair programming 10

matlab


Copy code

```
function index = binary_search(arr, low, high, x)
    % Check base case
    if high >= low
        mid = floor((high + low) / 2);
        if arr[mid] == x
            index = mid;
            return;
        elseif arr[mid] > x
            index = binary_search(arr, low, mid - 1, x);
            return;
        else
            index = binary_search(arr, mid + 1, high, x);
            return;
        end
    end
    else
        index = -1;
    end
end

% Test array
arr = [2, 3, 4, 10, 40];
x = 10;

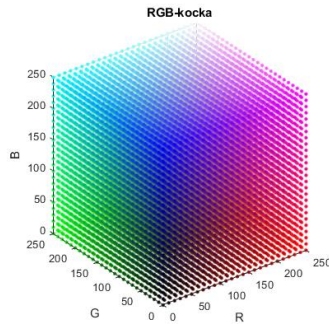
% Function call
result = binary_search(arr, 1, length(arr), x);

if result ~= -1
    fprintf('Element is present at index %d\n', result);
else
    fprintf('Element is not present in array\n');
end
```



Pair programming 11

• Kóddokumentum



Browser tabs: SZTAKI Szótár: szota..., Wikipedia, Fontos!, Kutatás, Érdekes, Könyv, ujság, Magyar, helyesiras.mta.hu, iLovePDF, Dijnet, PressReader

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Untitled.ipynb

Code

Notebook No Kernel

Browser tabs: Csepi, Kéktúra, Arduino, Kiegy, Python, S, W, B, R, G, K, B, AI courses free

Fájlok: Intro.ipynb, Lorenz.zip, sqLite.ipynb, Untitled.ipynb

Fájlok: D:\Downloads\Matlab_markdown_test2.html

Műszaki Dokumentáció: RGB-Kocka

Áttekintés

Ez a MATLAB kód egy 3D-s ábrát hoz létre, amely megjeleníti az összes lehetséges RGB színek kombinációját egy kockában. Minden pont a kockában egy egyedi színek kombinációját képvisel az RGB (piros, zöld, kék) színtérben.

Kód részletezése

- figure:** Ez a parancs egy új ábrablakot nyit.
- plot3(0,0,0):** Ez létrehoz egy háromdimenziós plotot. Itt még nem rajzol semmit, csak inicializálja a 3D plotot.
- hold on:** Ez a parancs lehetővé teszi, hogy több grafikai elemet is rajzoljunk az aktuális ábrára anélkül, hogy az előzőek törlődnének.
- Három egymásba ágyazott **for** ciklus (R, G, B változókkal): Ezek a ciklusok végigterátnak az összes lehetséges RGB színek kombinációját. Az **R**, **G**, és **B** változók értékei 0-tól 255-ig terjednek, 10-es lépésekkel növekedve.
- plot3(R,G,B,'.', 'Color',1./255*[R G B], 'MarkerSize',10):** Ez a parancs egy pontot rajzol az adott (R, G, B) koordinátákra. A pont színe az aktuális RGB értékek alapján van beállítva, a $1/255 \cdot [R \ G \ B]$ kifejezés normalizálja az RGB értékeket 0 és 1 közé. A **MarkerSize** beállítással a pont méretét növeljük.
- grid on, axis equal:** A **grid on** parancs rácsot ad a plotra a jobb orientáció érdekében. Az **axis equal** beállítás biztosítja, hogy az összes tengely azonos skálázású legyen, így a kocka arányos marad.
- title('RGB-kocka');** Ez ad címet az ábrának.
- xlabel('R'),ylabel('G'),zlabel('B');** Ezek a parancsok címkéket adnak az X, Y és Z tengelyekhez, megjelölve őket mint R (piros), G (zöld) és B (kék) tengelyeket.

Kimenet

Az elkészült ábra egy kockát fog mutatni, ahol minden pont egy különböző RGB színek kombinációját képvisel. A kocka tengelyei az egyes színek intenzitását mutatják (0-tól 255-ig), lehetővé téve a színek kombinációjának vizuális áttekintését.



A ChatGPT még az oktatásban

BA

You

Az alábbi kódot írtuk a sin-függvény érték

%% sin(x) számítása

```
x = 40*pi/180;
```

```
y = 0;
```

```
for i = 0 : 5
```

```
    dy = ((-1)^i)/factorial(2*i+1)*x^(2*i+1);
```

```
    y = y + dy;
```

```
end
```

```
y
```

Ellenőrizd, hogy a kód helyes és válaszolj

Hozz létre 5 olyan kérdést, aminek segít



ChatGPT

A kód helyes, és az Taylor-sorozat segítségével közelíti a `sin(x)` függvény értékét, ahol `x` radiánban van megadva. Az iterációs ciklus 5 iteráción keresztül fut, és minden iterációban hozzáadja a következő tagot a sorozatból az `y` változóhoz.

5 ellenőrző kérdés a megértéshez:

1. Miért használjuk a `(-1)^i` kifejezést a sorozatban?
2. Mi a célja a `factorial(2*i+1)` kifejezésnek a kódban?
3. Milyen előnyei vannak annak, hogy a `sin(x)` értékét Taylor-sorozat segítségével közelítjük?
4. Hogyan változna a közelítés pontossága, ha növelnénk az iterációk számát a ciklusban?
5. Miért szükséges az `x` értékét radiánban megadni a `sin(x)` kiszámításához?

A ChatGPT még az oktatásban 2

3. Alapvető geometriai transzformációk

A geometriai transzformációk létezését vizsgáljuk:

- az érintett pontokat megadjuk a változatlan koordináta-rendszerben

BA

You



Alapvető geometriai transzformációk
Document

A feltöltött dokumentum az alábbi kérdésekre
Készíts olyan 2 feleletválasztós kérdést
(betűkkel azonosítva) van és
egyértelműen!



ChatGPT

Alapján a dokumentumra, amire
van két feleletválasztós kérdés
helyes. A helyes válaszok egy

1. Melyik két állítás igaz az eltolás geometriai transzformációjára?

- A) Az eltolás során a koordináták értékei változatlanok maradnak.
- B) Az eltolás során a koordináta-rendszer mozgatásával érjük el a transzformációt.
- C) Az eltolás a pontok helyzetének megváltoztatásával történik a változatlan koordináta-rendszerben. (Helyes)
- D) Az eltolás mindig az X tengely mentén történik.
- E) Az eltolás hatására a pontok relatív helyzete megváltozik a koordináta-rendszerhez képest. (Helyes)

2. Melyik két állítás igaz a síkbeli forgatásra?

- A) A forgatás során a pontok koordinátái nem változnak.
- B) A forgatás a koordináta-tengelyek körül, meghatározott szögekkel történik. (Helyes)
- C) A forgatás csak az X tengely körül lehetséges.
- D) A síkbeli forgatásnál a forgatási szöget az origóból nézve az óramutató járásával ellentétes irányban tekintjük pozitívnak. (Helyes)
- E) A forgatás során a pontok távolsága az origótól változik.

„milyen módon lehet a ChatGPT-t bevonni az egyetemi oktatásba?”



ChatGPT

A ChatGPT-t számos módon lehet bevonni az egyetemi oktatásba, íme néhány példa:

1. **Támogató eszközként:** A ChatGPT-t használhatják a diákok és oktatók is kérdések megválaszolására, fogalmak magyarázatára vagy akár példák generálására.
2. **Tananyag-fejlesztés:** A modell segíthet tananyagok, például gyakorlati feladatok, tesztek vagy összefoglalók készítésében.
3. **Nyelvi támogatás:** A ChatGPT segíthet a nyelvi készségek fejlesztésében, különösen a külföldi nyelvek tanulásában, ahol a diákoknak kell szerezniük a nyelvi kompetenciákat.
4. **Szimulációk és szerepjátékok:** A ChatGPT-t lehet használni, ahol a diákoknak interaktív környezetben kell kommunikálniuk vagy döntéseket hozniuk.
5. **Kutatási támogatás:** A ChatGPT segíthet az ötletgenerálásban, irodalomkutatásban vagy akár a kutatási tervek kialakításában.
6. **Beszélgetések és viták:** A modellt beszélgetések és viták katalizátoraként is lehet használni, például etikai vagy filozófiai témákban, ahol a különböző nézőpontok megvitatása fontos.
7. **Feladatok automatizált értékelése:** Automatizált visszajelzés vagy értékelés biztosítása diákok frásos munkáira.

Fontos megjegyezni, hogy bár a ChatGPT hasznos kiegészítője lehet az oktatásnak, nem helyettesítheti az oktatók személyes hozzájárulását és a kritikai gondolkodást igénylő interaktív tanulási folyamatokat.





Köszönöm a figyelmet!