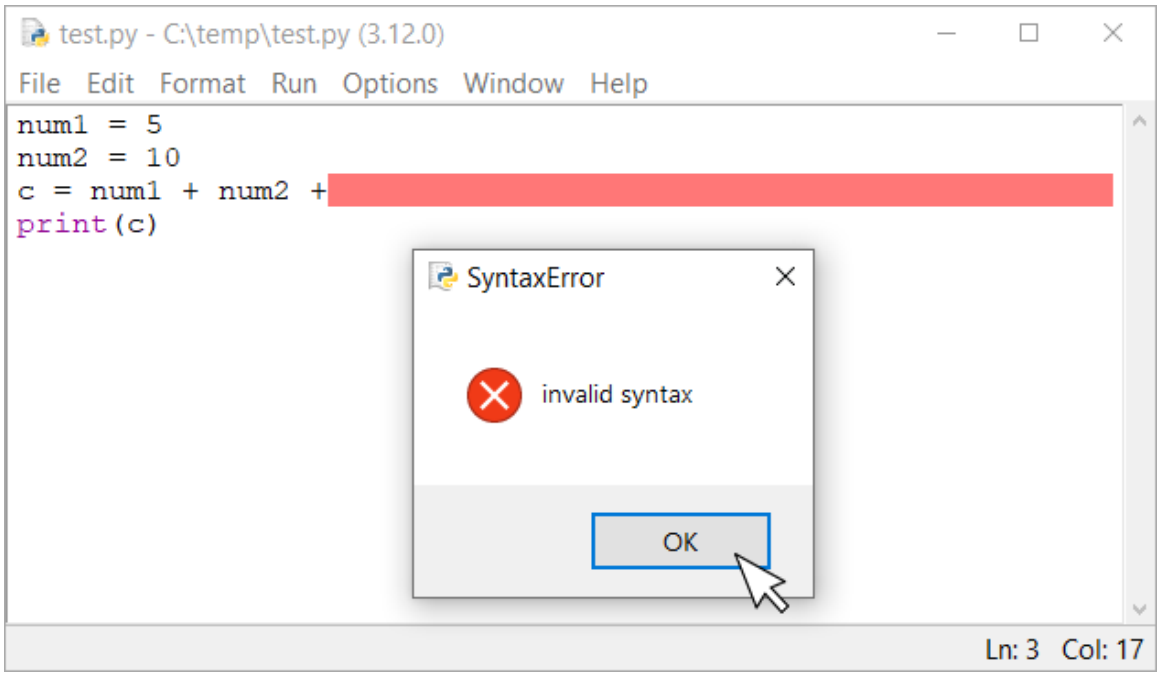
06. prosinac 2025.

UPOTREBA DEBUGGERA U PYTHON PROGRAMSKOM JEZIKU



Snježana Šavorić, 4B2

Što je debugger?

Debugger je alat koji nama programerima olakšava život. Pomaže pri **pronalasku** i samom **ispravljanju pogrešaka u kodu**.

Mogućnosti debugger-a

Debugger ima sposobnost:

- Zaustavljanja programa na određenoj liniji
- Prikazivanja trenutnih vrijednosti varijabli
- Koračanja kroz kod
- Otkrivanja grešaka
- I tako dalje…

Programiranje i pogreške u kodu

U programiranju razlikujemo 3 vrste pogreška:

1. **Sintaktičke**
2. **Logičke greške: program se i dalje izvršava bez ikakvih runtime ili sintaktičkih pogrešaka, ali rezultat nije onakav kakav želimo. Zbog toga se ovaj tip grešaka ne može otkriti običnim debuggerom**
3. **Greške pri izvršavanju programa**

Sigurno se pitate: "*Kako debugger pomaže pri rješavanju ovakvih pogrešaka?*"

Ja ću danas demonstrirati na 2 vlastita programa.

Program #1 – vlastita funkcija za računanje šifre

2

1. GREŠKA PRI IZVOĐENJU PROGRAMA (linija 15)

- *Nastaju **tijekom izvođenja** programa.*
- *Python može "pročitati" kod, ali kad dođe do problematične linije, program se sruši.*

program.py > main

```
1   def izracunajSifru(B):
2       umnozak = 1
3       for znamenka in str(B):
4           umnozak *= int(znamenka)
5
6       if umnozak % 2 == 0:
7           sifra = umnozak + 1
8       else:
9           sifra = umnozak + 2
10
11      return sifra
12
13  def main():
14      B = int(input("Unesi broj B: "))
15      sifra = izracunajsifru(B)
16      print("Jankova šifra je:", sifra)
17  if __name__ == "__main__":
18      main()
19
```

Kako debugger odgovara na ovakvu pogrešku?

```
15      sifra = izracunajsifru(B) B = 5
```

```
Exception has occurred: NameError ×
name 'izracunajsifru' is not defined
  File "C:\Users\angel\New folder\program.py", line 15, in main
    sifra = izracunajsifru(B)
            ^^^^^^^^^^^^^^
  File "C:\Users\angel\New folder\program.py", line 18, in <module>
    main()
    ~~~~^^
NameError: name 'izracunajsifru' is not defined
```

- *Debugger je **istaknuo liniju** u kojoj se događa greška*
- *Debugger je **naglasio o kojoj liniji koda se radi** te je liniju i **podcrtao***
- *Debugger je **rekao** o kojem **tipu pogrešaka** se radi*

2. Sintaktička greška (linija 8)

- *Nastaju **kada kršimo pravila** programskog jezika.*

program.py > izracunajSifru

```python
def izracunajSifru(B):
    umnozak = 1
    for znamenka in str(B):
        umnozak *= int(znamenka)

    if umnozak % 2 == 0:
        sifra = umnozak + 1
    else
        sifra = umnozak + 2

    return sifra

def main():
    B = int(input("Unesi broj B: "))
    sifra = izracunajSifru(B)
    print("Jankova šifra je:", sifra)
if __name__ == "__main__":
    main()

```

Kako debugger odgovara na ovakvu pogrešku?

```
File                                          line 8
  else
      ^
SyntaxError: expected ':'
```

- *Debugger je **naglasio** o kojoj liniji koda se radi*
- *Debugger je **naglasio što se trebalo učiniti***
- *Debugger je **rekao** o kojem **tipu pogrešaka** se radi*

REZULTAT = SAVRŠENO ISPRAVAN PROGRAM:

ctions... ogram.py > ...

```python
def izracunajSifru(B):
    umnozak = 1
    for znamenka in str(B):
        umnozak *= int(znamenka)

    if umnozak % 2 == 0:
        sifra = umnozak + 1
    else:
        sifra = umnozak + 2

    return sifra

def main():
    B = int(input("Unesi broj B: "))
    sifra = izracunajSifru(B)
    print("Jankova šifra je:", sifra)
if __name__ == "__main__":
    main()

```

```
Unesi broj B: 21
Jankova šifra je: 3
```

Program #2 – klasa

5

1. GREŠKA PRI IZVOĐENJU PROGRAMA (linija 11)

- *Nastaju **tijekom izvođenja** programa.*
- *Python može "pročitati" kod, ali kad dođe do problematične linije, program se sruši.*

```
1   class Zaposlenik:
2       def __init__(self, staz, placa, odradeniDani):
3           self.staz = staz
4           self.placa = placa
5           self.odradeniDani = odradeniDani
6       def zapocniSmjenu(self):
7           print("Zaposlenik je zapoceo smjenu")
8           self.odradeniDani += 1
9   def main():
10      zaposlenik1 = Zaposlenik(20, 2000, 10)
11      zaposlenik1.zapocnismjenu()
12      print(zaposlenik1.odradeniDani)
13
14  if __name__ == "__main__":
15      main()
16
```

Kako debugger odgovara na ovakvu pogrešku?

```
1   class Zaposlenik:
2       def __init__(self, staz, placa, odradeniDani):
3           self.staz = staz
4           self.placa = placa
5           self.odradeniDani = odradeniDani
6       def zapocniSmjenu(self):
7           print("Zaposlenik je zapoceo smjenu")
8           self.odradeniDani += 1
9   def main():
10      zaposlenik1 = Zaposlenik(20, 2000, 10) zaposlenik1 = <__main__.Zaposlenik object a
11      zaposlenik1.zapocnismjenu()
```

Exception has occurred: AttributeError ×

```
'Zaposlenik' object has no attribute 'zapocnismjenu'
  File "C:\Users\angel\New folder\program.py", line 11, in main
    zaposlenik1.zapocnismjenu()
    ^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\angel\New folder\program.py", line 15, in <module>
    main()
    ~~~~^^
AttributeError: 'Zaposlenik' object has no attribute 'zapocnismjenu'
```

- *Debugger je **naglasio o kojoj liniji koda se radi** te je liniju i **podcrtao***
- *Debugger je **rekao** o kojem **tipu pogrešaka** se radi*
- *Debugger je **istaknuo** liniju u kojoj se događa greška*

2. Sintaktička greška (linija 11)

- *Nastaju **kada kršimo pravila** programskog jezika.*

```
1   class Zaposlenik:
2       def __init__(self, staz, placa, odradeniDani):
3           self.staz = staz
4           self.placa = placa
5           self.odradeniDani = odradeniDani
6       def zapocniSmjenu(self):
7           print("Zaposlenik je zapoceo smjenu")
8           self.odradeniDani += 1
9   def main():
10      zaposlenik1 = Zaposlenik(20, 2000, 10)
11      zaposlenik1.zapocniSmjenu(
12      print(zaposlenik1.odradeniDani)
13
14  if __name__ == "__main__":
15      main()
16
```

Kako debugger odgovara na ovakvu pogrešku?

```
File                                    , line 11
  zaposlenik1.zapocniSmjenu(
                            ^
SyntaxError: '(' was never closed
```

- *Debugger je **naglasio** o kojoj liniji koda se radi*
- *Debugger je **naglasio što se trebalo učiniti***
- *Debugger je **rekao** o kojem **tipu pogrešaka** se radi*

REZULTAT = SAVRŠENO ISPRAVAN PROGRAM:

program.py > ...

```python
class Zaposlenik:
    def __init__(self, staz, placa, odradeniDani):
        self.staz = staz
        self.placa = placa
        self.odradeniDani = odradeniDani
    def zapocniSmjenu(self):
        print("Zaposlenik je zapoceo smjenu")
        self.odradeniDani += 1
def main():
    zaposlenik1 = Zaposlenik(20, 2000, 10)
    zaposlenik1.zapocniSmjenu()
    print(zaposlenik1.odradeniDani)

if __name__ == "__main__":
    main()

```

```
Zaposlenik je zapoceo smjenu
11
```

Zaključak: u zlatnom dobu umjetne inteligencije lako se zaboravi da se problemi u programiranju mogu riješiti i uz alate poput debuggera. Ovaj rad to dokazuje.