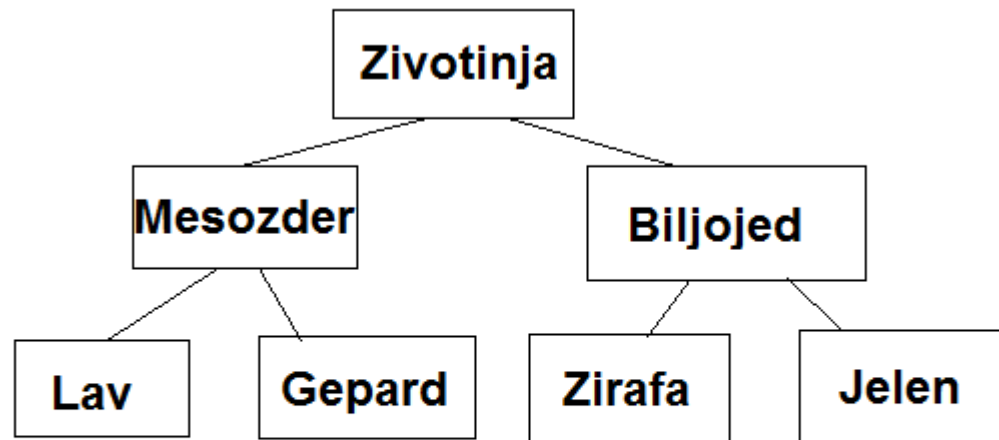


Klase i objekti



Objekti



Klase

- Klasa opisuje objekat
- Kod definisanja klase se koristi rezervisana reč **class**
- Primer:

```
class Klasa:  
    pass
```

*pass označava da klasa nema nikakve podatke

Klase - nasledjivanje

- Klasa može da nasledi drugu klasu i pri tome preuzima za sebe sve podatke nadklase.
- Klasa koja nasledjuje se naziva potklasa ili klasa dete a klasu koju nasledjuju se naziva nadklasa ili klasa roditelj
- Jedna klasa može da nasledi samo jednu klasu
- Jednu klasu može da nasledi neograničen broj klasa

Klase - nasledjivanje

- **Primer:**

```
class Zivotinja:
```

```
    pass
```

```
class Mesojder(Zivotinja):
```

```
    pass
```

```
class Lav(Mesojder):
```

```
    pass
```

Razlika izmedju klase i objekta

- Lav je klasa
- Lav Pera koji ima 35 godina, mužjak je, i živi u Beogradskom zoološkom vrtu je objekat

Pera = Lav()

Podaci klase

- Klasa može imati svoje podatke koji se nazivaju atributi klase
- Primer:

```
class Klasa:
```

```
    promenljiva = 1234
```

Funkcije klasa

```
class Klasa:  
    def funkcija():  
        print("Funkcija!")
```


Funkcije i nasljedjivanje

- Pri nasljedjivanju, potklasa preuzima sve podatke i funkcije nadklase.

- Primer:

```
class Mesojder:
```

```
    def kretanje(self):
```

```
        print("Trčim")
```

```
class Gepard(Mesojder):
```

```
    pass
```

```
gopard = Gepard()
```

```
gopard.kretanje()
```

Klase i nasljedjivanje

- Moguće je menjati nasledjene funkcije

Primer:

```
class Gepard(Mesozder):
```

```
    def kretanje:
```

```
        print("trcim veoma brzo")
```

Init funkcija

```
class A:  
    def __init__(self):  
        self.x = 'Hello'  
    def method_a(self, foo):  
        print("%s %s" % (self.x, string))
```

Opseg promenljivih

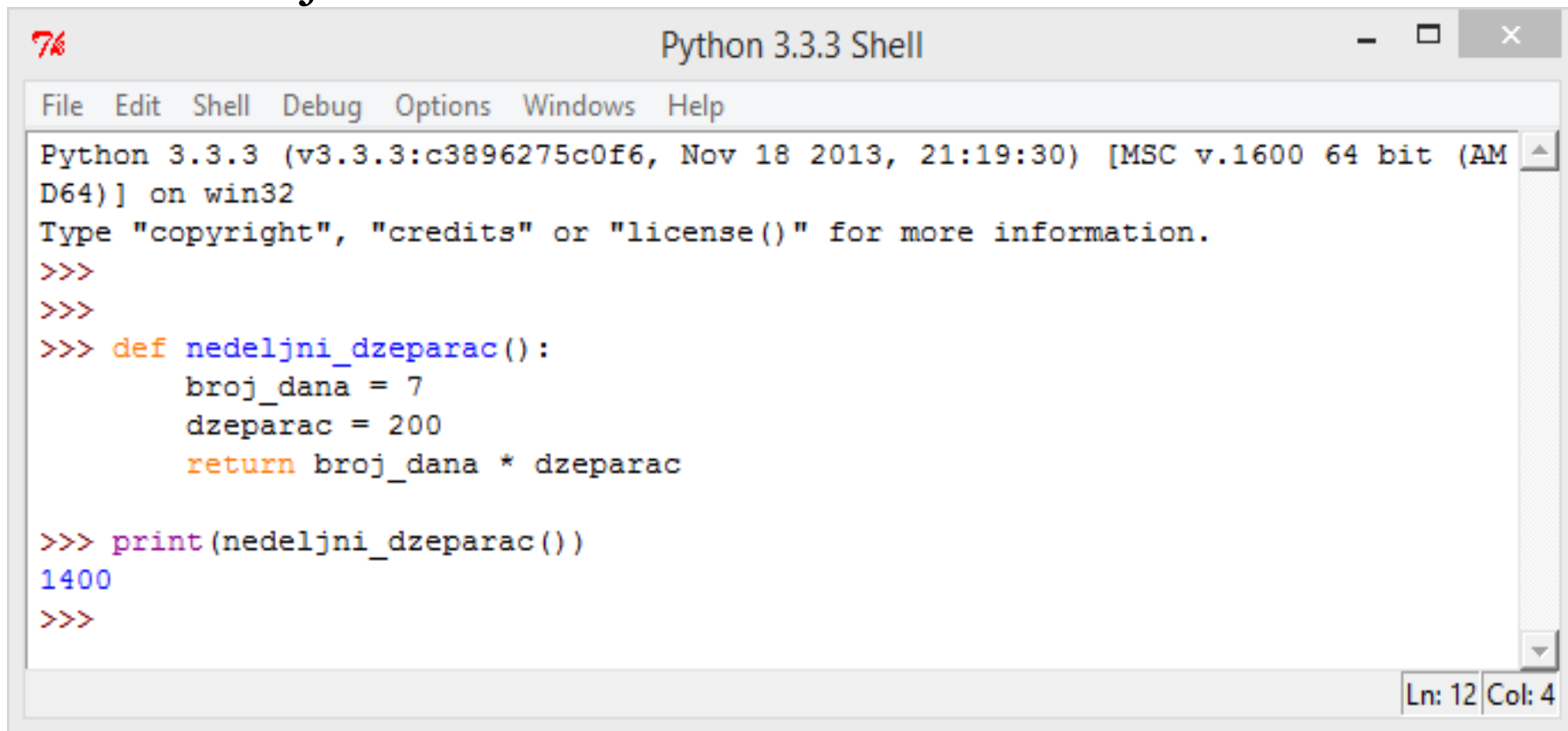


Promenjive i opseg promenjivih

- Promenjiva koja se nalazi unutar tela funkcije ne može se koristiti ponovo nakon što je funkcija završila sa radom zato što promenjiva postoji samo u okviru te funkcije.
- U programiranju to se naziva opseg (vidljivost) promenljive

Opseg promenljivih

- Napravimo funkciju nedeljni_dzeparac koja množi dve promenjive (broj_dana i dzeparac) i vraća rezultat množenja.

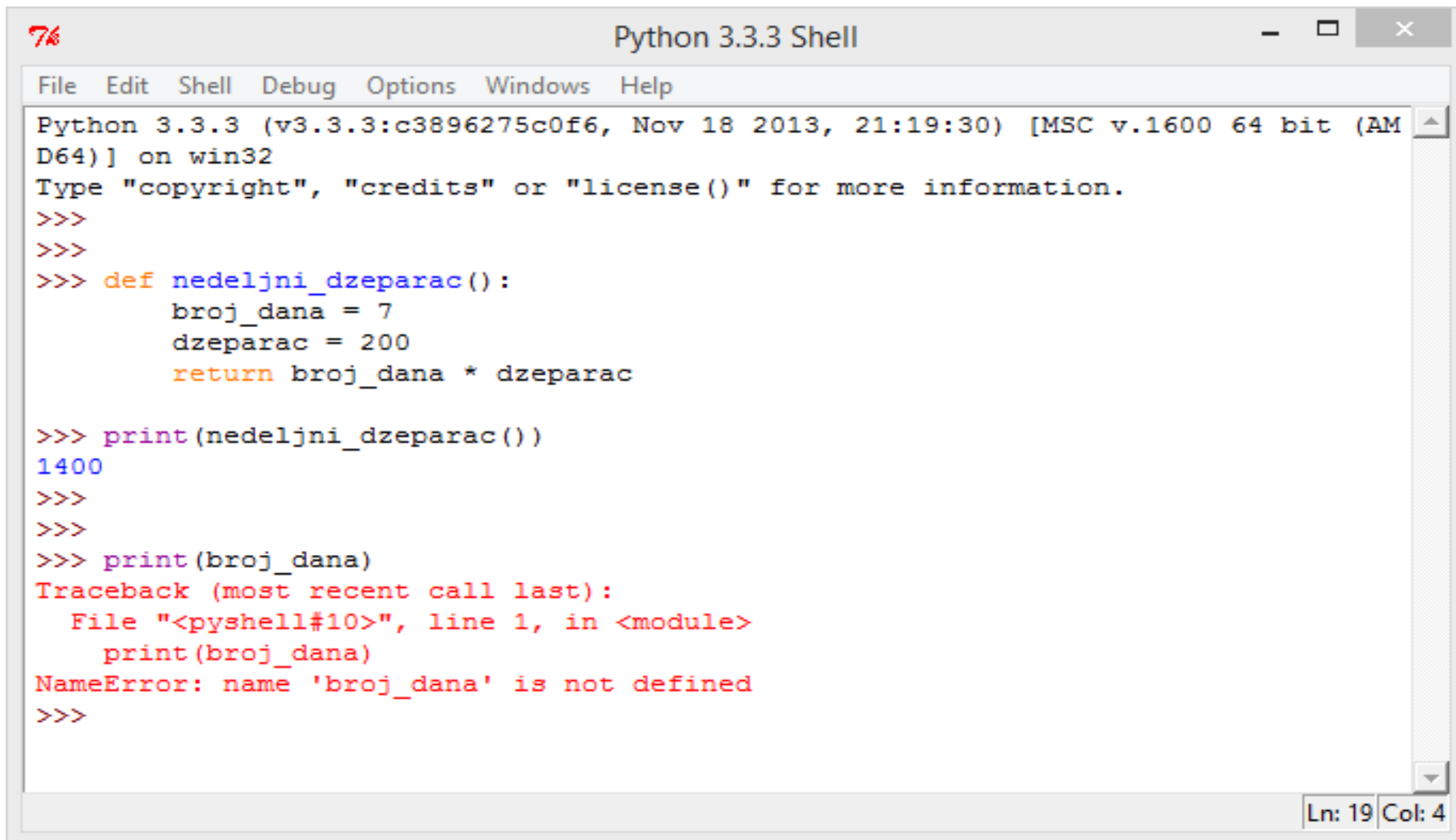


```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> def nedeljni_dzeparac():
    broj_dana = 7
    dzeparac = 200
    return broj_dana * dzeparac

>>> print(nedeljni_dzeparac())
1400
>>>
```

Ln: 12 Col: 4

Opseg promenljivih



```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>> def nedeljni_dzeparac():
    broj_dana = 7
    dzeparac = 200
    return broj_dana * dzeparac

>>> print(nedeljni_dzeparac())
1400
>>>
>>>
>>> print(broj_dana)
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    print(broj_dana)
NameError: name 'broj_dana' is not defined
>>>
```

Ln: 19 Col: 4

Opseg promenljivih

- ako promenjivu definišemo van naše funkcije, ona ima drugačiji opseg.
- Na primer, ako promenjivu **broj_dana** definišemo van funkcije **nedeljni_dzeparac** i ako pokušamo da oštampamo tu promenjivu dobićemo njenu vrednost.
- Tako deklarisanе promenjive se nazivaju **globalne promenjive** i one su vidljive u okviru jedne ili više funkcija, kao i izvan njih.

Opseg promenljivih

```
Python 3.3.3 Shell
File Edit Shell Debug Options Windows Help
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:19:30) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
>>>
>>> broj_dana = 7
>>> def nedeljni_dzeparac():
>>>     dzeparac = 200
>>>     return broj_dana * dzeparac
>>> print(nedeljni_dzeparac())
1400
>>>
>>> print(broj_dana)
7
>>>
```

Ln: 17 Col: 4