Programare I
Curs 6,7
Depanare. Aserțiuni. Excepții.
Fișiere text, CSV, JSON. I/O cu fișiere. [1]

Botescu Mihai
mihai.botescu00@e-uvt.ro

May 2021

```
   1:
try:
    a = int(input())
except ValueError as e:
    print(e)
   2:
try:
    a = int(input())
except ValueError as e:
    print(e)
   3:
try:
    a = int(input())
except ValueError as e:
    print(e)
   4:
try:
    a = int([1,2,3])
except TypeError as e:
    print(f'Type error: {e}')
finally:
    print('Am terminat.')
   5:
try:
    a = int(2)
except TypeError as e:
    print(f'Type error: {e}')
finally:
    print('Am terminat.')
   6:
try:
    print(x)
except SyntaxError as e:
    print(f'Syntax error: {e}')
finally:
    print('Am terminat.')
   7:
try:
    print(x)
except NameError as e:
```

```
    print(f'Syntax error: {e}')
finally:
    print('Am terminat.')
   8:
try:
    raise NotImplementedError()
except  NotImplementedError as e:
    print(f'Not implemented error: {e}')
finally:
    print('Mai implementeaza')
   9:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
finally:
    print('gata')
  10:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
finally:
    print('gata')
  11:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
finally:
    print('gata')
  12:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
```

```
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
finally:
    print('gata')
  13:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  14:
try:
    a=int(input())
    b=int(input())
    print(a/b)
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  15:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  16: from math import sqrt
  17:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
```

```
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  18:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  19:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  20:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  21:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
```

```
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  22:
try:
    a=int(input())
    print(sqrt(a))
except  TypeError  as e1:
    print(f'Type error: {e1}')
except ValueError as e2:
    print(f'Value error: {e2}')
except ZeroDivisionError as e3:
    print(f'Zero Division Error : {e3}')
except ArithmeticError as e4:
    print(f'Arithmetic Error : {e4}')
finally:
    print('gata')
  23:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError)  as (e1,e2,e3,e4):
    print('Eroare')
  24:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  25:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  26:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  27:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  28:
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  29:
```

```
try:
    a=int(input())
    print(sqrt(a))
except  (TypeError,ValueError,ZeroDivisionError,ArithmeticError):
    print('Eroare')
  30:
with open('fisier.txt','r') as f:
    print(f.read())
    print(f.readline())
    print(f.readlines())
  31:
with open('fisier.txt','r') as f:
    #print(f.read())
    print(f.readline())
    print(f.readlines())
  32:
def citeste_linie_cu_linie(f):
    with open(f,'r'):
        linie = f.readline()
        while linie is not None:
            print(linie)
            linie = f.readline()
  33: citeste_linie_cu_linie('fisier.txt')
  34: citeste_linie_cu_linie(f)
  35:
def citeste_linie_cu_linie(file):
    with open(file,'r') as f:
        linie = f.readline()
        while linie is not None:
            print(linie)
            linie = f.readline()
  36: citeste_linie_cu_linie('fisier.txt')
  37:
def citeste_linie_cu_linie(file):
    with open(file,'r') as f:
        linie = f.readline()
        while linie is not None and len(linie) > 0:
            print(linie)
            linie = f.readline()
  38: citeste_linie_cu_linie('fisier.txt')
  39: citeste_linie_cu_linie('lipsum_1.txt')
  40:
def citeste_deodata(file):
    with open(file,'r') as f:
        linie = f.read()
        print(linie)
  41: citeste_deodata('lipsum_1.txt')
  42: citeste_deodata('lipsum_1.txt')
  43: 0%20
  44: 0%20
  45: 0%2020
  46: %history -g
```