

Εισαγωγή στον Προγραμματισμό Η/Υ

Διδάσκοντες: Ι. Πολενάκης, Κ. Σκιαδόπουλος, Δ. Μουρατίδου

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο



8. Εξαιρέσεις και Αρχεία

► Τι είναι οι Εξαιρέσεις (Exceptions);

- Σφάλματα που εμφανίζονται κατά την εκτέλεση
- Διακόπτουν τη φυσιολογική ροή του προγράμματος
- Μας επιτρέπουν να χειριστούμε προβλήματα με ελεγχόμενο τρόπο

► Γιατί χρειαζόμαστε Εξαιρέσεις;

- Προγράμματα πιο ανθεκτικά
- Καλύτερη εμπειρία χρήστη
- Αποφυγή «κρashaρίσματος» εφαρμογών

8. Εξαιρέσεις και Αρχεία

▶ Παράδειγμα Σφάλματος

```
x = int(input("Δώσε αριθμό: "))  
print(10 / x)
```

Τι μπορεί να πάει στραβά;

8. Εξαιρέσεις και Αρχεία

► Τι είναι το `try / except`

- `try`: κώδικας που μπορεί να αποτύχει
- `except`: τι κάνουμε αν συμβεί σφάλμα

8. Εξαιρέσεις και Αρχεία

► Τι είναι το try / except

- **try**: κώδικας που μπορεί να αποτύχει
- **except**: τι κάνουμε αν συμβεί σφάλμα

Βασική Δομή try / except

try:

```
x = int(input("Δώσε αριθμό: "))  
print(10 / x)
```

except:

```
print("Κάτι πήγε στραβά")
```

8. Εξαιρέσεις και Αρχεία

➤ Συγκεκριμένες Εξαιρέσεις

Καλύτερη πρακτική:

```
except ValueError:
```

```
except ZeroDivisionError:
```

Παράδειγμα με Πολλαπλά except

```
try:
```

```
    x = int(input("Δώσε αριθμό: "))
```

```
    print(10 / x)
```

```
except ValueError:
```

```
    print("Δεν έδωσες ακέραιο")
```

```
except ZeroDivisionError:
```

```
    print("Διαίρεση με το μηδέν")
```

8. Εξαιρέσεις και Αρχεία

➤ Σειρά Εξαιρέσεων

- Η Python ελέγχει από πάνω προς τα κάτω
- Το πρώτο except που ταιριάζει εκτελείται

➤ Το finally

- ❑ Εκτελείται πάντα
- ❑ Χρήσιμο για καθαρισμό πόρων

```
try:
    f = open("data.txt")
    print(f.read())
except FileNotFoundError:
    print("Το αρχείο δεν βρέθηκε")
finally:
    print("Τέλος εκτέλεσης")
```

8. Εξαιρέσεις και Αρχεία

➤ Συνηθισμένες Εξαιρέσεις

- ❑ ValueError
- ❑ TypeError
- ❑ FileNotFoundError
- ❑ ZeroDivisionError

➤ **except as e**

```
try:  
    int("abc")  
except ValueError as e:  
    print(e)
```

8. Εξαιρέσεις και Αρχεία

➤ Συνηθισμένες Εξαιρέσεις

- ☐ ValueError
- ☐ TypeError
- ☐ FileNotFoundError
- ☐ ZeroDivisionError

➤ Γενικό vs Ειδικό except

... γράφουμε **except ValueError:** και όχι σκέτο **except:**

8. Εξαιρέσεις και Αρχεία

- Χρήση Εξαιρέσεων σε Συναρτήσεις

```
def divide(a, b):  
    try:  
        return a / b  
    except ZeroDivisionError:  
        return None
```

- raise – Δημιουργία Εξαίρεσης

```
if age < 0:  
    raise ValueError("Η ηλικία δεν μπορεί να είναι αρνητική")
```

Πότε χρησιμοποιούμε raise

- Έλεγχος εγκυρότητας δεδομένων
- Καθαρός διαχωρισμός λογικής & ελέγχου

8. Εξαιρέσεις και Αρχεία

➤ Εξαιρέσεις και Αρχεία

```
try:
    with open("students.txt") as f:
        print(f.read())
except FileNotFoundError:
    print("Αρχείο δεν βρέθηκε")
```

8. Εξαιρέσεις και Αρχεία

➤ Εξαιρέσεις και Αρχεία

```
try:
```

```
    with open("students.txt") as f:
```

```
        print(f.read())
```

```
except FileNotFoundError:
```

```
    print("Αρχείο δεν βρέθηκε")
```

➤ Τι είναι τα Αρχεία;

- Αποθήκευση δεδομένων μόνιμα
- Text files (.txt, .csv)
- Ανάγνωση / Εγγραφή από πρόγραμμα Python

8. Εξαιρέσεις και Αρχεία

➤ Άνοιγμα Αρχείου

```
f = open("data.txt", "r")  
content = f.read()  
f.close()
```

➤ Γιατί χρησιμοποιούμε **with**

- Αυτόματο κλείσιμο αρχείου
- Αποφυγή σφαλμάτων αν ξεχαστεί το close

```
with open("data.txt", "r") as f:  
    content = f.read()
```

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Ανάγνωση Αρχείων

- `read()` : όλο το περιεχόμενο
- `readline()` : μια γραμμή
- `readlines()` : λίστα γραμμών

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Εγγραφή Αρχείων

```
with open("students.txt", "w") as f:  
    f.write("Maria,20\n")  
    f.write("Nikos,22\n")
```

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Προσθήκη σε Αρχείο

```
with open("students.txt", "a") as f:  
    f.write("Alex,21\n")
```

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Ανάγνωση CSV

```
import csv

with open('students.csv', newline='') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)
```

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Εγγραφή CSV

```
import csv  
  
with open('students.csv', 'w', newline='') as csvfile:  
    writer = csv.writer(csvfile)  
    writer.writerow(['Name', 'Age'])  
    writer.writerow(['Maria', 20])
```

8. Εξαιρέσεις και Αρχεία

➤ Modes Αρχείων

Mode	Περιγραφή
r	Ανάγνωση
w	Εγγραφή (σβήνει περιεχόμενο)
a	Προσθήκη
rb / wb	Binary mode

➤ Binary Αρχεία

```
with open('image.jpg', 'rb') as f:  
    data = f.read()
```

8. Εξαιρέσεις και Αρχεία

► Διαχείριση Σφαλμάτων Αρχείων

```
try:
    with open("data.txt") as f:
        print(f.read())
except FileNotFoundError:
    print("Το αρχείο δεν βρέθηκε")
```

❑ Συνδυασμός with + try / except

- Ασφαλής χειρισμός αρχείων
- Αντιμετώπιση σφαλμάτων

- Αρχεία = μόνιμη αποθήκευση
- Με with = ασφαλές κλείσιμο
- Εξαιρέσεις = ασφαλής χειρισμός λαθών

8. Εξαιρέσεις και Αρχεία

➤ Άλλες δυνατότητες...

- Άνοιγμα αρχείου με διαφορετική κωδικοποίηση

```
with open('data.txt', 'r', encoding='utf-8') as f:  
    content = f.read()
```

- Διαχείριση μεγάλων αρχείων - Ανάγνωση γραμμή-γραμμή για οικονομία μνήμης

```
with open('large.txt') as f:  
    for line in f:  
        process(line)
```

8. Εξαιρέσεις και Αρχεία

➤ Άλλες δυνατότητες...

■ Αντιγραφή Αρχείων

```
import shutil  
shutil.copy('source.txt', 'dest.txt')
```

■ Διαγραφή Αρχείων

```
import os  
os.remove('file.txt')
```

■ Έλεγχος αν υπάρχει αρχείο

```
import os  
if os.path.exists('file.txt'):  
    print('Υπάρχει')
```

8. Εξαιρέσεις και Αρχεία

➤ Άλλες δυνατότητες...

■ Αντιγραφή Αρχείων

```
import shutil  
shutil.copy('source.txt', 'dest.txt')
```

■ Διαγραφή Αρχείων

```
import os  
os.remove('file.txt')
```

■ Έλεγχος αν υπάρχει αρχείο

```
import os  
if os.path.exists('file.txt'):  
    print('Υπάρχει')
```



8. Εξαιρέσεις και Αρχεία

- Αναπτύξτε ένα πρόγραμμα που θα επιτρέπει:
 1. Δημιουργία αρχείου με φοιτητές
 2. Προσθήκη νέων
 3. Ανάγνωση και εμφάνιση όλων
 4. Χειρισμός σφαλμάτων

8. Εξαιρέσεις και Αρχεία

➤ Αναπτύξτε ένα πρόγραμμα που θα επιτρέπει:

1. Δημιουργία αρχείου με φοιτητές
2. Προσθήκη νέων
3. Ανάγνωση και εμφάνιση όλων
4. Χειρισμός σφαλμάτων

```
1  import os
2  FILENAME = "students.txt"
3  def add_student(name, age):
4      """Προσθέτει έναν φοιτητή στο αρχείο"""
5      try:
6          with open(FILENAME, "a") as f:
7              f.write(f"{name},{age}\n")
8              print(f"Ο φοιτητής {name} προστέθηκε.")
9      except Exception as e:
10         print(f"Σφάλμα κατά την προσθήκη: {e}")
11
12  def show_students():
13      """Διαβάζει και εμφανίζει όλους τους φοιτητές"""
14      if not os.path.exists(FILENAME):
15          print("Δεν υπάρχει αρχείο φοιτητών.")
16          return
17      try:
18          with open(FILENAME, "r") as f:
19              lines = f.readlines()
20              if not lines:
21                  print("Δεν υπάρχουν εγγραφές.")
22                  return
23              print("Λίστα Φοιτητών:")
24              for line in lines:
25                  name, age = line.strip().split(",")
26                  print(f"- {name}, {age} ετών")
27      except Exception as e:
28          print(f"Σφάλμα κατά την ανάγνωση: {e}")
```

8. Εξαιρέσεις και Αρχεία

➤ Αναπτύξτε ένα πρόγραμμα που θα επιτρέπει:

1. Δημιουργία αρχείου με φοιτητές
2. Προσθήκη νέων
3. Ανάγνωση και εμφάνιση όλων
4. Χειρισμός σφαλμάτων

```
30 def main():
31     while True:
32         print("\n--- Μενού ---")
33         print("1. Προσθήκη φοιτητή")
34         print("2. Εμφάνιση όλων των φοιτητών")
35         print("3. Έξοδος")
36         choice = input("Επιλέξτε μια επιλογή: ")
37         if choice == "1":
38             name = input("Όνομα φοιτητή: ")
39             age_input = input("Ηλικία φοιτητή: ")
40             try:
41                 age = int(age_input)
42                 add_student(name, age)
43             except ValueError:
44                 print("Λάθος ηλικία! Δώστε ακέραιο αριθμό.")
45         elif choice == "2":
46             show_students()
47         elif choice == "3":
48             print("Έξοδος προγράμματος.")
49             break
50         else:
51             print("Μη έγκυρη επιλογή. Προσπαθήστε ξανά.")
52     if __name__ == "__main__":
53         main()
```

```
1 import os
2 FILENAME = "students.txt"
3 def add_student(name, age):
4     """Προσθέτει έναν φοιτητή στο αρχείο"""
5     try:
6         with open(FILENAME, "a") as f:
7             f.write(f"{name},{age}\n")
8             print(f"Ο φοιτητής {name} προστέθηκε.")
9     except Exception as e:
10        print(f"Σφάλμα κατά την προσθήκη: {e}")
11
12 def show_students():
13     """Διαβάζει και εμφανίζει όλους τους φοιτητές"""
14     if not os.path.exists(FILENAME):
15         print("Δεν υπάρχει αρχείο φοιτητών.")
16         return
17     try:
18         with open(FILENAME, "r") as f:
19             lines = f.readlines()
20             if not lines:
21                 print("Δεν υπάρχουν εγγραφές.")
22                 return
23             print("Λίστα Φοιτητών:")
24             for line in lines:
25                 name, age = line.strip().split(",")
26                 print(f"- {name}, {age} ετών")
27     except Exception as e:
28         print(f"Σφάλμα κατά την ανάγνωση: {e}")
```

1. Χρησιμοποιεί αρχείο students.txt για αποθήκευση.
2. Παρέχει μενού για: Προσθήκη νέου φοιτητή, Εμφάνιση όλων των φοιτητών, Έξοδος
3. Κάνει έλεγχο σφαλμάτων: Μη έγκυρη ηλικία (ValueError), Σφάλματα κατά το άνοιγμα/ανάγνωση/εγγραφή αρχείου
4. Χρησιμοποιεί with για ασφαλές άνοιγμα αρχείου.