

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Askisi4:

1)γ) Δεν υπάρχει abstract μέθοδο χωρίς υλοποίηση στην κλάση **MyTester**, οπότε δεν ήταν απαραίτητος ο χαρακτηρισμός της ως abstract κλάση .

Περιπτώσεις:

{}

Αποτελέσματα:

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married, with 3 children.

Με τον παραπάνω κώδικα εκτυπώσαμε τα χαρακτηριστικά των αντικειμένων p1 και mp1 αντίστοιχα.

2)Περιπτώσεις:

{}

Αποτελέσματα:

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married.

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married.

Το πλεονέκτημα του καινούργιου κώδικα έναντι του προηγούμενου είναι ότι ο καινούργιος έχει δυνατότητα επαναχρησιμοποίησης. Ωστόσο, υπήρχε ανάγκη για γενίκευση των μηνυμάτων οπότε απουσιάζει η δυνατότητα εκτύπωσης του αριθμού των παιδιών, η οποία ολοκληρώνει την πρόταση.

3)Περιπτώσεις:

{}

Αποτελέσματα:

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married,

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married, with 3 children.

4)δ) Περιπτώσεις:

{}

Αποτελέσματα:

Paul Kings is 22 years old, gets a 1200.0 Euros salary and is married,

Betty Tront is 31 years old, gets a 980.5 Euros salary and is married, with 3 children.

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Δεν έχουν τροποποιηθεί κατάλληλα οι μέθοδοι printInfo, οπότε δεν εμφανίζεται η τιμή της μεταβλητής sex στο μήνυμα.

4)ε) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Paul Kings is 22 years old and male, gets a 1200.0 Euros salary and is married,

Betty Tront is 31 years old and female, gets a 980.5 Euros salary and is married, with 3 children.

5)β) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Paul Kings is 22 years old and male, gets a 1200.0 Euros salary and is not married.

Betty Tront is 31 years old and female, gets a 980.5 Euros salary and is not married. with 3 children.

Is mp1 married?: false

Χρησιμοποιώντας την μέθοδο setMarried(false) στο mp1, αλλάξαμε τη τιμή που είχε ο constructor της κλάσης MarriedPerson για την μεταβλητή married, αλλάζοντας ουσιαστικά την “ειδοποιό διαφορά” των κλάσεων MarriedPerson και Person και το βασικό χαρακτηριστικό που χαρακτήριζε το στιγμιοτύπο mp1 ως αντικείμενο της κλάσης MarriedPerson και όχι της Person.

5)γ) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Paul Kings is 22 years old and male, gets a 1200.0 Euros salary and is not married.

Betty Tront is 31 years old and female, gets a 980.5 Euros salary and is married, with 3 children.

Is mp1 married?: true

6) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Betty Tront is 31 years old and female, gets a 980.5 Euros salary and is married, with 3 children.

Kirk Tront is 31 years old and male, gets a 2080.0 Euros salary and is married, with 2 children.

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Sonia Tront is 31 years old and female, gets a 600.0 Euros salary and is married, with 0 children.

Betty Tront is 31 years old and female, gets a 3060.5 Euros salary and is married, with 3 children.

Betty Tront is 31 years old and female, gets a 3060.5 Euros salary and is married, with 3 children.

Betty Tront is 31 years old and female, gets a 2080.0 Euros salary and is married, with 3 children.

Η μέθοδος `setSalary` της κλάσης `MarriedPerson` επικαλύπτει τη μέθοδο `setSalary` της κλάσης `Person`.

7)α) Δεν είναι δυνατή η προσθήκη του τρίτου στιγμιότυπου στον πίνακα `mpArray` διότι ο πίνακας έχει δημιουργηθεί στατικά με 2 μόνο θέσεις και δεν μπορεί να μεταβληθεί αυθαίρετα.

7)β) Η δομή `ArrayList` είναι δυναμική οπότε έχει μεταβαλλόμενο, κατά τη διάρκεια εκτέλεσης του προγράμματος, μέγεθος σε αντίθεση με τη `Array`, η οποία είναι στατική, με σταθερό μέγεθος.

7)γ) Η κλάση `Iterator` στην `java.util` δεν μπορεί να μετατραπεί σε μία ειδικότερη κλάση, την κλάση `MarriedPerson`, για το λόγο αυτό και εμφανίζει εξαίρεση ο μεταγλωττιστής.

7)ε) Ο `Iterator` είναι αναγκαίος στην περίπτωση όπου θέλουμε να αφαιρέσουμε στοιχεία από τον πίνακα ή την λίστα, γενικότερα να επηρεάσουμε τον πίνακα ή την λίστα χωρίς να εμφανιστεί εξαίρεση κατά την διάρκεια “σάρωσης” της λίστας ή του πίνακα.

Askisi5:

1)β) `unreported exception java.io.IOException; must be caught or declared to be thrown.`

1)γ) Στην νέα δομή της `read`, γίνεται χρήση της `try`, η οποία εμφανίζει μήνυμα στην περίπτωση όπου εμφανιστεί το `IOException`, αντιμετωπίζοντας το κατά κάποιο τρόπο, ολοκληρώνοντας τη μεταγλώττιση.

2) Περιπτώσεις:

‘3’ ‘2.2’

Αποτελέσματα:

Dwste enan akeraio:

3

Dwste enan pragmatiko:

2.2

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

i=3 f=2.2

‘zzz’ ‘9’

```
java.lang.NumberFormatException: For input string: "zzz"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.parseInt(Integer.java:615)
    at IO_Tester.readInt(IO_Tester.java:12)
    at IO_Tester.main(IO_Tester.java:58)
```

3) Περιπτώσεις:

Αποτελέσματα:

‘zzz’ ‘9’

Dwste enan akeraio:

zzz

Exception: java.lang.NumberFormatException: For input string: "zzz"

Returned value: -1

Dwste enan pragmatiko:

9

i=-1 f=9.0

Ο νέος κώδικας για την μέθοδο readInt(), αντιμετωπίζει και εξαίρεση διαφορετική μορφής εισόδου έναντι της αναμενόμενης και επιστρέφει το -1, ως τιμή εισόδου, επιτρέποντας έτσι την συνέχιση της εκτέλεσης του κώδικα.

Περιπτώσεις:

Αποτελέσματα:

‘zzz’ ‘γγγ’

Dwste enan akeraio:

zzz

Exception: java.lang.NumberFormatException: For input string: "zzz"

Returned value: -1

Dwste enan pragmatiko:

γγγ

Exception: java.lang.NumberFormatException: For input string: "γγγ"

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Returned Value: -1.0

i=-1 f=-1.0

4) Περιπτώσεις:

'4' '10' 'xch' 'True'

Αποτελέσματα:

Dwste enan akeraio:

4

Dwste enan float:

10

Dwste ena string:

xch

Dwste mia boolean:

True

i=4 f=10.0 s=xch b=true

'number' '4.5f' 'some_text' 'true_again'

Dwste enan akeraio:

number

Exception: java.lang.NumberFormatException: For input string: "number"

Returned value: -1

Dwste enan float:

4.5f

Dwste ena string:

some_text

Dwste mia boolean:

true_again

i=-1 f=4.5 s=some_text b=false

'-1' 'ff' '34' '12'

Dwste enan akeraio:

-1

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Dwste enan float:

ff

Exception: java.lang.NumberFormatException: For input string: "ff"

Returned Value: -1.0

Dwste ena string:

34

Dwste mia boolean:

12

i=-1 f=-1.0 s=34 b=false

Στην περίπτωση '4' '10' 'xch' 'True' , το πρόγραμμα δέχεται ως ακέραιο το 4, ως float το 10, ως String το 'xch' και ως Boolean την τιμή True.

Στην περίπτωση 'number' '4.5f' 'some_text' 'true_again' , το πρόγραμμα δέχεται ως ακέραιο το String 'number', οπότε εμφανίζεται ένα εξαίρεση διαφορετική μορφής εισόδου έναντι της αναμενόμενης και εκτελείται ο αντίστοιχος κώδικας για την αντιμετώπισή του, επιστέφοντας την τιμή -1. Ως float το 4.5, ως String το 'some_text' και ως Boolean το String 'true_again', το οποία μεταφράζεται ως false, καθώς το String δεν αντιστοιχεί σε Boolean τιμή.

Στην περίπτωση '-1' 'ff' '34' '12' , το πρόγραμμα δέχεται ως ακέραιο το -1, ως float το String 'ff', οπότε εμφανίζεται ένα εξαίρεση διαφορετική μορφής εισόδου έναντι της αναμενόμενης και εκτελείται ο αντίστοιχος κώδικας για την αντιμετώπισή του, επιστέφοντας την τιμή -1.0. Ως String το '34' και ως Boolean τον ακέραιο 12, ο οποίος μεταφράζεται ως false, καθώς ο ακέραιος δεν αντιστοιχεί σε Boolean τιμή.

5)α) unreported exception java.io.FileNotFoundException; must be caught or declared to be thrown.

5)β) Εξαιτίας του καινούργιου κώδικα ,η μεταγλώττιση ολοκληρώνεται, διότι λαμβάνεται υπόψη η πιθανότητα εμφάνισης εξαίρεσης.

5)γ)Η μεταγλώττιση δεν ολοκληρώθηκε διότι δεν λαμβάνεται υπόψη η πιθανότητα απουσίας εισόδου για τις μεθόδους writeInt(), writeFloat(), writeBoolean() και close().

5)δ) Περιπτώσεις:

'10' '.25e-2' 'test.log' 'tRUe'

Αποτελέσματα:

Dwste enan akeraio:

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

10

Dwste enan float:

25e-2

Dwste ena string:

test,log

Dwste mia boolean:

tRUE

i=10 f=0.25 s=test,log b=true

5)ε) Δημιουργήθηκε ένα αρχείο που περιέχει τα εξής δεδομένα >€ []. Δημιουργείται ένα στιγμιότυπο fil της κλάσης FileOutputStream, με όρισμα το όνομα του αρχείου το String " test,log", δημιουργώντας το αρχείο με όνομα test.log. Έπειτα δημιουργείται ένα στιγμιότυπο buf της κλάσης BufferedOutputStream με όρισμα το στιγμιότυπο fil, επιτρέποντας έτσι την εγγραφή byte σε έναν buffer χωρίς να απαιτείται μια κλήση στο υποκείμενο σύστημα για κάθε γραπτό byte και τελικά δημιουργείται ένα στιγμιότυπο dat της κλάσης DataOutputStream με όρισμα το στιγμιότυπο buf, το οποίο επιτρέπει στο πρόγραμμα να γράψει "πρωτόγονους" τύπους δεδομένων στο αρχείο test,log. Έπειτα καλείται η μέθοδος writeInt(), η οποία γράφει τον ακέραιο i στο stream με 4 bytes. Αντιστοίχα με την σειρά τους η writeFloat() τον float αριθμό f και η writeBoolean() την Boolean μεταβλητή b. Τέλος καλείται η μέθοδος close(), η οποία κλείνει το αρχείο και απελευθερώνει τους πόρους του συστήματος που εμπλέκονταν με αυτό το stream.

6)β) Περιπτώσεις:

'10' '25e-2' 'test.log' 'tRUE'

Αποτελέσματα:

Dwste enan akeraio:

10

Dwste enan float:

25e-2

Dwste ena string:

test.log

Dwste mia boolean:

tRUE

i=10 f=0.25 s=test.log b=true

i=20 f=0.0625 s=test.log b=false

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Αρχικά δημιουργείται ένα στιγμιότυπο `fil` της κλάσης `FileInputStream` με όρισμα το όνομα του αρχείου `"test.log"`, το οποίο λαμβάνει τα `byte` από το αρχείο `test.log`. Έπειτα, δημιουργείται ένα στιγμιότυπο `buf` της κλάσης `BufferedInputStream` με όρισμα το στιγμιότυπο `fil`, γεμίζοντας τον `buffer` με τα `byte` του αρχείου και στη συνέχεια δημιουργείται ένα στιγμιότυπο `dat` της κλάσης `DataInputStream` με όρισμα το στιγμιότυπο `buf`, το οποίο επιτρέπει την ανάγνωση "πρωτόγονων" τύπων δεδομένων από τον `buffer`. Καλείται η μέθοδος `readInt()`, η οποία διαβάζει το αρχείο `test.log` και επιστρέφει τον ακέραιο που είχε εγγραφεί σε αυτό, δηλαδή το 10, το οποίο με την σειρά του προστίθεται στο `i` οπότε και εκτυπώνεται το 20. Ομοίως καλείται η μέθοδος `readFloat()`, οπότε το 0.25 πολλαπλασιάζεται με το `f`, εκτυπώνοντας το 0.0625 και η μέθοδος `readBoolean()`, οπότε εφαρμόζεται το λογικό NOT στη επιστρεφόμενη τιμή `true`, εκτυπώνοντας την τιμή `false`. Τελικά καλείται η μέθοδος `close()`, η οποία κλείνει το αρχείο και απελευθερώνει τους πόρους του συστήματος που εμπλέκονταν με αυτό το `stream` και εκτυπώνονται οι νέες τιμές.

6)δ) Περιεχόμενο του αρχείου: 200.0625test.logtest.logfalse

Ο νέος κώδικας γράφει χαρακτήρες αντί για `byte` στο αρχείο `test.log` και χρησιμοποιώντας την μέθοδο `write()` εγγράφει τις τιμές όλων των μεταβλητών στο αρχείο.

Askisi6:

1)β) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 7

Sub: 3

Mul: 10

Div: 2

1)γ) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 5

Sub: 5

Mul: 0

`java.lang.ArithmeticException: / by zero`

`at Exception_Tester.printResults(Exception_Tester.java:15)`

`at Exception_Tester.main(Exception_Tester.java:9)`

Εμφανίζεται αριθμητικό εξάιρεση διότι δεν ορίζεται διαίρεση με το 0.

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

2) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 5

Sub: 5

Mul: 0

java.lang.ArithmeticException: / by zero

Χάρης το νέο κώδικα λαμβάνεται υπόψη η πιθανότητα εμφάνισης αριθμητικού λάθους λόγω απόπειρας διαίρεσης με το 0 και εκτυπώνεται το όνομα του εξαίρεσης.

3) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 5

Sub: 5

Mul: 0

java.lang.ArithmeticException: / by zero

Ο καινούργιος κώδικας επέκτεινε την δυνατότητα αντιμετώπισης της εξαίρεσης σε όλη την μέθοδο printResults(), έναντι μόνο της διαίρεσης.

4) Στην περίπτωση αυτή γίνεται το throw μιας εξαίρεσης, χωρίς όμως να γίνεται το catch της.

5)α) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 5

Sub: 5

Mul: 0

java.lang.ArithmeticException: / by zero

The numbers are: 5 0

5)β) Περιπτώσεις:

```
{}
```

Αποτελέσματα:

Add: 7

Sub: 3

Mul: 10

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

Div: 2

The numbers are: 5 2

5)γ) Το block της finally εκτελείται ανεξάρτητα αν θα εμφανιστεί εξαίρεση ή όχι στο try-catch block.

6)α) Περιπτώσεις:

{}

Αποτελέσματα:

Add: 7

Sub: 3

Mul: 10

Div: 2

java.lang.ArithmeticException

The numbers are: 5 2

Η νέα γραμμή κώδικα κάνει throw ένα δεύτερο arithmetic exception.

6)β) Η μέθοδος printStackTrace() εμφανίζει που ακριβώς εμφανίστηκαν οι εξαιρέσεις ανεξάρτητα αν έγιναν catch ή όχι, οπότε εμφανίζει 2 εξαιρέσεις μία στη μέθοδο printResults() και μία στην main.

7)α) Το μόνο που άλλαξε στα αποτελέσματα είναι το είδος της εξαίρεσης, από το γενικό είδος java.lang.ArithmeticException άλλαξε στο ειδικότερο DivideByZeroException.

7)β) Περιπτώσεις:

{}

Αποτελέσματα:

Add: 7

Sub: 3

Mul: 10

Div: 2

DivideByZeroException: The denominator cannot be zero.

The numbers are: 5 2

DivideByZeroException: The denominator cannot be zero.

at Exception_Tester.printResults(Exception_Tester.java:29)

Αναφορά για το 2ο σετ εργαστηριακών ασκήσεων JAVA

at Exception_Tester.main(Exception_Tester.java:11)

7)γ) Κατά την διάρκεια εκτέλεσης του προγράμματος, εμφανίζονται 2 εξαιρέσεις ίδιου τύπου, εκ των οποίων μόνο η μία γίνεται catch. Ωστόσο, δεν λαμβάνεται υπόψη η πιθανότητα διαίρεσης με το 0, οπότε και εμφανίζεται μία ακόμη εξαίρεση.

7)δ) Περιπτώσεις:

{}

Αποτελέσματα:

Add: 5

Sub: 5

Mul: 0

DivideByZeroException: The denominator cannot be zero.

The numbers are: 5 0

DivideByZeroException: The denominator cannot be zero.

at Exception_Tester.printResults(Exception_Tester.java:32)

at Exception_Tester.main(Exception_Tester.java:11)

Ο καινούργιος κώδικας αντιμετωπίζει μόνο το μία από τις δύο εξαιρέσεις, αυτή που δημιουργείται λόγω της διαίρεσης με το 0, αδυνατώντας να αντιμετωπίσει την δεύτερη.