Άσκηση 4

1.Η κλάση ορίζεται ως abstract διότι δεν έχει κατασκευαστή. Δεν χρειάζεται να οριστεί abstract καθώς δεν έχει abstract μεθόδους.

Η MyTester εκτυπώνει:

> 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.

2.Το πλεονέκτημα αυτού του κώδικα έναντι του προηγουμένου είναι ότι πλέον δεν θα χρειάζεται να γράφουμε όλες τις εντολές για κάθε Person για το οποίο θέλουμε να εκτυπώσουμε τις πληροφορίες αρά υπάρχει συντομία στον χρόνο συγγραφής του κώδικα και στην έκταση του.
Πλέον εκτυπώνεται:
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.

Παρατηρούμε όμως ότι δεν εκτυπώνεται η πληροφορία για το ποσά παιδιά έχει ο κάθε Person το οποίο είναι αναμενόμενο καθώς η πληροφορία για τα παιδιά βρίσκεται στην κλάση MarriedPerson.

3.Εκτυπώνεται:

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)Δεν παρατηρείται αλλαγή διότι στις printInfo δεν έγινε αλλαγή ώστε να εκτυπώνονται τα φύλλα.

ε)

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

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

5.β)

Paul Kings is Male is 22 years old, gets a 1200.0 Euros salary and is not married.
Betty Tront is Female is 31 years old, gets a 980.5 Euros salary and is married with 3 children.
Is mp1 married?: false

γ)

Paul Kings is Male is 22 years old, gets a 1200.0 Euros salary and is not married.
Betty Tront is Female is 31 years old, gets a 980.5 Euros salary and is married with 3 children.
Is mp1 married?: true

6)

Betty Tront is Female is 31 years old, gets a 980.5 Euros salary and is married with 3 children.
Kirk Tront is Male is 31 years old, gets a 2080.0 Euros salary and is married with 2 children.
Sonia Tront is Female is 31 years old, gets a 600.0 Euros salary and is married with no children.
Betty Tront is Female is 31 years old, gets a 3060.5 Euros salary and is married with 3 children.
Betty Tront is Female is 31 years old, gets a 3060.5 Euros salary and is married with 3 children.
Betty Tront is Female is 31 years old, gets a 2080.0 Euros salary and is married with 3 children.

Η mp1.setSalary() με όρισμα mp2 προσθέτει τους μισθούς της Betty και του Kirk καθώς είναι διαφορετικά φύλλα.

Η mp1.setSalary() με όρισμα mp3 δεν κάνει τίποτα διότι η Betty και η Sonia είναι το ίδιο φύλλο.

Τέλος, η mp1.setSalary() με όρισμα mp2.getSalary() είναι η setSalary() της Person με όρισμα float οπότε βάζει τον μισθό του mp2 στο mp1.

7)

α)Όχι δεν μπορώ να προσθέσω εκ των υστέρων και το τρίτο στιγμιότυπο διότι ο πίνακας έχει προκαθορισμένο μέγεθος.

Betty Tront is Female is 31 years old, gets a 980.5 Euros salary and is married with 3 children.
Kirk Tront is Male is 31 years old, gets a 2080.0 Euros salary and is married with 2 children.

β)

Το πλεονέκτημα του ArrayList έναντι του Array είναι ότι το πρώτο δεν έχει προκαθορισμένο μέγεθος και έτσι μπορούμε να προσθέτουμε συνέχεια νέα αντικείμενα σε αυτό.

γ)

Το πρόβλημα του μεταγλωττιστή είναι ότι δεν έχουμε ορίσει τύπο στο iter.

Άσκηση 5

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

Αυτό το σφάλμα βγαίνει γιατί δεν χειριζόμαστε την περίπτωση IOException.

(γ) Αυτή τη φορά ο μεταφραστής δεν βγάζει σφάλμα γιατί τώρα χρησιμοποιούμε try-catch

2)1η εκτέλεση:

Dwste enan akeraio:
3
Dwste enan pragmatiko:
2.2
i=3 f=2.2
2η εκτέλεση:

Dwste enan akeraio:
Zzz
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:10)
at IO_Tester.main(IO_Tester.java:60)

3) 1η εκτέλεση:

Dwste enan akeraio:
zzz
Exception: java.lang.NumberFormatException: For input string: "zzz"
Returned value: -1
Dwste enan pragmatiko:
9
i=-1 f=9.0

2η εκτέλεση:

Dwste enan akeraio:
zzz
Exception: java.lang.NumberFormatException: For input string: "zzz"
Returned value: -1
Dwste enan pragmatiko:
yyy
Exception: java.lang.NumberFormatException: For input string: "yyy"
Returned value: -1
i=-1 f=-1.0

4) 1η εκτέλεση:

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

Επιστρέφετε ο integer που εισήγαμε (4).
Επιστρέφετε ο float που εισήγαμε(10).
Επιστρέφετε το String που εισήγαμε(xch).
Επιστρέφετε η boolean που βάλαμε(true)

2η εκτέλεση:

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

Επιστρέφετε -1 επειδή η είσοδος μας δεν είναι τύπου integer.
Επιστρέφετε ο float που εισήγαμε(4,5). (Το f δεν χρειάζεται αφού δηλώνει ότι ο αριθμός είναι float, ενώ ο αριθμός είναι ήδη float.)
Επιστρέφετε το String που εισήγαμε(some_text).
Επιστρέφετε false γιατί δεν εισήγαμε κάποια τιμή που να παριστάνει την boolean τιμή true.

3η εκτέλεση:

Dwste enan akeraio:
-1
Dwste enan float:
ff
Exception: java.lang.NumberFormatException: For input string: "ff"
Returned value: -1
Dwste ena string:

34
Dwste mia boolean:
12
i=-1 f=-1.0 s=34 b=false

Επιστρέφετε ο integer που εισήγαμε(-1).
Επιστρέφετε -1 καθώς αυτό που εισήγαμε δεν είναι τύπου float.
Επιστρέφετε το String που εισήγαμε(34).
Επιστρέφετε false αφού εισήγαμε false.

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

(β) Όχι γιατί αυτήν την φορά χειριζόμαστε το FileNotFoundException με μία try-catch.

(γ) Σφάλμα: unreported exception java.io.IOException; must be caught or declared to be thrown

(δ) Όχι γιατί αυτή την φορά χειριζόμαστε το IOException με μία try-catch.

(ε) Εκτέλεση:

Dwste enan akeraio:
10
Dwste enan float:
.25e-2
Dwste ena string:
test.log

Dwste mia boolean:
tRUe
i=10 f=0.0025 s=test.log b=true

(ε) Περιεχόμενα test.log:

;#X

Αυτός ο κώδικας χρησιμοποιείτε για να αποθηκεύσει (γράψει) πληροφορίες σε ένα αρχείο για να τις διαβάσει το πρόγραμμα, δεν προορίζεται για δική μας κατανόηση οπότε δεν μπορούμε να καταλάβουμε τι λέει.

6)(β) Εκτέλεση:

Dwste enan akeraio:
10
Dwste enan float:
.25e-2
Dwste ena string:
test.log
Dwste mia boolean:
tRUe
i=10 f=0.0025 s=test.log b=true
i=20 f=6.25E-6 s=test.log b=false

Ο καινούργιος κώδικας διαβάζει τα δεδομένα που έχουμε αποθηκεύσει από το (5.ε) και:

Προσθέτει στον integer αυτόν που έχει αποθηκευτεί στο test.log.
Πολλαπλασιάζει τον float με αυτόν που έχει αποθηκευτεί στο test.log.
Το String παραμένει ίδιο.

Κάνει τον Boolean αντίθετο από αυτό που έχει αποθηκευτεί στο test.log(Δηλαδή true -> false και false -> true)

(δ) Εκτέλεση:

Dwste enan akeraio:
10
Dwste enan float:
.25e-2
Dwste ena string:
test.log
Dwste mia boolean:
tRUe
i=10 f=0.0025 s=test.log b=true
i=20 f=6.25E-6 s=test.log b=false

Περιεχόμενα test.log:
206.25E-6test.logtest.logfalse

Τα ορίσματα στη write() είναι ένα string, ένας integer και ένας integer. Αυτή η μέθοδος γραφεί σε ένα αρχείο το string ξεκινώντας από τον χαρακτήρα με στη θέση του πρώτου integer (0) μέχρι και την θέση του δευτέρου. Το test.log εμφανίζεται δυο φορές λόγω του concat που ενώνει δυο string στην περίπτωση μας είναι το ίδιο και τέλος με το bw.close() κλείνει το stream αφού το αδειάσει.

Άσκηση 6

1)β)

Add: 7
Sub: 3
Mul: 10
Div: 2

γ)

Add: 5
Sub: 5
Mul: 0
java.lang.ArithmeticException: / by zero
at Exception_Tester.printResults(Exception_Tester.java:14)
at Exception_Tester.main(Exception_Tester.java:8)

Βγαίνει σφάλμα-εξαίρεση διότι δεν μπορούμε να διαιρέσουμε με το 0

2)

Add: 5
Sub: 5
Mul: 0
java.lang.ArithmeticException: / by zero

Αυτή τη φορά χειριζόμαστε την εξαίρεση με μια try-catch. Έτσι αντί για σφάλμα μας εκτυπώνει ποια εξαίρεση προέκυψε αφού έχουμε βάλει την εντολή ae.toString που μετατρέπει στο ρεύμα πληροφορίας για την εξαίρεση σε αναγνώσιμο αλφαριθμητικό.

3)

Add: 5
Sub: 5
Mul: 0
java.lang.ArithmeticException: / by zero

Πλέον χειριζόμαστε την εξαίρεση μέσα στον κώδικα που καλεί την printResults. Αυτό δεν αλλάζει τα αποτελέσματα αλλά υπάρχει κίνδυνος σε πιθανή επαναχρησιμοποίηση της printResults καθώς θα πρέπει να ξαναχρησιμοποιήσουμε την try-catch.

4)Όπως ήταν αναμενόμενο στο αποτέλεσμα ξανά έχουμε το σφάλμα-αποτέλεσμα. Έτσι πρέπει να χειριστούμε τις εξαιρέσεις όπως προηγουμένως.

5)α)

Add: 5
Sub: 5
Mul: 0
java.lang.ArithmeticException: / by zero
The numbers are: 5 0

β)

Add: 7
Sub: 3
Mul: 10
Div: 2
The numbers are: 5 2

γ) Τα αποτελέσματα στην 2 πλέον είναι κανονικά διότι δεν υπάρχει πλέον διαίρεση με 0. Το finally σε μια try-catch κάνει τις εντολές στο σώμα του ανεξαρτήτως του αν υπήρξε εξαίρεση, έτσι, σε κάθε περίπτωση εκτυπώνεται το μήνυμα μας.

6) α)

Add: 7
Sub: 3
Mul: 10
Div: 2
java.lang.ArithmeticException
The numbers are: 5 2

Καλείται η throw η οποία «πετάει» μια εξαίρεση στην περίπτωση μας java.lang.ArithmeticException.

β)

Add: 7
Sub: 3
Mul: 10
Div: 2
java.lang.ArithmeticException
The numbers are: 5 2
java.lang.ArithmeticException

```
        at Exception_Tester.printResults(Exception_Tester.java:25)
        at Exception_Tester.main(Exception_Tester.java:9)
        at __SHELL12.run(__SHELL12.java:6)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native  Method)
        at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at bluej.runtime.ExecServer$3.run(ExecServer.java:743)
```

Μέσω της printStackTrace() εντοπίζεται η μέθοδος που προκάλεσε την εξαίρεση.

7)α)

```
 Add: 7
Sub: 3
Mul: 10
Div: 2
Exception_Tester$DivideByZeroException
The numbers are: 5 2
Exception_Tester$DivideByZeroException
        at Exception_Tester.printResults(Exception_Tester.java:25)
        at Exception_Tester.main(Exception_Tester.java:9)
        at __SHELL13.run(__SHELL13.java:6)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native  Method)
        at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
        at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
43)
        at java.lang.reflect.Method.invoke(Method.java:498)
        at bluej.runtime.ExecServer$3.run(ExecServer.java:743)
```

Με την νεα προσθήκη στον κώδικα πέρα από το μήνυμα για το ποια εξαίρεση εμφανίζεται, δεν έχουμε άλλες αλλαγές καθώς όπως αναφέρθηκε και πιο πριν η printStackTrace() εντοπίζει ποια μέθοδος προκάλεσε την εξαίρεση και αφού είναι στην ιδιά σειρά και στις δυο περιπτώσεις δεν υπάρχουν διάφορες στα αποτελέσματα.

β)

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:25)
at Exception_Tester.main(Exception_Tester.java:9)
at __SHELL14.run(__SHELL14.java:6)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:498)
at bluej.runtime.ExecServer$3.run(ExecServer.java:743)

γ)

Add: 5
Sub: 5
Mul: 0
The numbers are: 5 0
java.lang.ArithmeticException: / by zero
at Exception_Tester.printResults(Exception_Tester.java:24)
at Exception_Tester.main(Exception_Tester.java:9)

Παρόλο που έχουμε ορίσει το divideByZero exception δεν έχουμε λύσει την περίπτωση του Arithmetic exception.

δ)

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:28)
at Exception_Tester.main(Exception_Tester.java:9)
at __SHELL18.run(__SHELL18.java:6)

at sun. reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.lang.reflect.Method.invoke(Method.java:498)

at bluej.runtime.ExecServer$3.run(ExecServer.java:743)

Πλέον το πρόγραμμα τερματίζεται ασφαλώς χειρίζεται τις εξαιρέσεις και «πετάει» DivideByZero Exception.