

# Object Oriented Programming Course (CEID\_NNY106)

## Εργαστηριακή Άσκηση

### Gui based RPN Calculator

#### 1. Στόχος

Η Εργαστηριακή Άσκηση (ΕΑ) έχει στόχο την εξοικείωση με:

- Τη δόμηση προγράμματος από ένα σύνολο από συνιστώσες (**system integration**)
- Την αξιοποίηση κλάσεων τρίτου κατασκευαστή οι οποίες δίνονται με την μορφή βιβλιοθηκών τύπου jar
- Την ανάπτυξη κλάσεων με ζητούμενη δομή και συμπεριφορά
- Επέκταση λειτουργικότητας και βελτίωση ποιότητας πηγαίου κώδικα
- Την Incremental development τεχνική στην ανάπτυξη αντικειμενοστρεφούς εφαρμογής
- ένα πιο επαγγελματικό IDE όπως είναι το Eclipse.

#### RPN Calculator από συνιστώσες

Η ΕΑ βασίζεται στις Δραστηριότητες 6.5 – 6.10 του κεφαλαίου 6 (ενότητες 6.4-6.6) και στις Δραστηριότητες 8.1-8.8 του κεφαλαίου 8 του βιβλίου “Από τις Υπολογιστικές στις Κυβερνο-Φυσικές Διεργασίες και το ΙοΤ: Αντικείμενα και Υπηρεσίες”, Κ. Θραμπουλίδης, ISBN 978-960-418-961-8, 2022, ΕΚΔΟΣΕΙΣ Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε. <https://sites.google.com/view/fromcomputationalto cyber-physi/home>

#### 2. Αναμενόμενα αποτελέσματα

Με την ολοκλήρωση της ΕΑ θα έχετε μια λειτουργούσα έκδοση της RPN Calculator με γραφική διεπαφή η οποία θα επιτρέπει τον υπολογισμό τιμής εκφράσεων remote polish notation.

Το βασικότερο όμως θα έχετε εξοικειωθεί:

1. με τη διαδικασία του **system integration** με αξιοποίηση έτοιμων συνθετικών που σας δίνονται υπό την μορφή αρχείων τύπου jar σε συνδυασμό με την **incremental development** τεχνική, και,
2. με την **ανάπτυξη συνιστωσών που θα ικανοποιούν δεδομένες προδιαγραφές**

### 3. Οδηγίες εκτέλεσης ΕΑ

Η ΕΑ αποτελείται από τρεις (3) ασκήσεις:

**Άσκηση Νο 1:** Σύνθεση της RPN Calculator από έτοιμα συνθετικά (System Integration),

**Άσκηση Νο 2:** Ανάπτυξη συνθετικών της RPN Calculator engine, και

**Άσκηση Νο 3:** Ανάπτυξη Γραφικής διεπαφής.

**Η Άσκηση Νο 1 έχει ως στόχο** να σας εξασκήσει με το integration των συνθετικών ενός συστήματος. Θα σας εξοικειώσει με τη διαδικασία της σύνθεσης (integration) συστήματος και θα σας βοηθήσει στην εκτέλεση των iterations των δύο επόμενων ασκήσεων.

**Στην Άσκηση Νο 2** θα έχετε τη δυνατότητα να αναπτύξετε εσείς τα συνθετικά της engine της αριθμομηχανής, τα συνθετικά της δηλαδή εκτός αυτού της γραφικής διεπαφής. Θα δημιουργήσετε έτσι μία έκδοση της engine με τα δικά σας συνθετικά.

**Στην Άσκηση Νο 3** θα δημιουργήσετε και την δική σας γραφική διεπαφή.

Οι ασκήσεις 2 και 3 υλοποιούνται με βάση την incremental development τεχνική. Κάθε ένα iteration είτε προσθέτει επιπλέον λειτουργικότητα στο αποτέλεσμα του προηγούμενου iteration είτε βελτιώνει την ποιότητα του κώδικα του προηγούμενου iteration αξιοποιώντας καλύτερα την αντικειμενοστρεφή προσέγγιση. Για τον λόγο αυτό **για κάθε iteration θα χρησιμοποιείτε ως βάση τον κώδικα που αναπτύξατε σε προηγούμενο iteration.**

**Μόνο όταν έχετε ολοκληρώσει επιτυχώς ένα iteration προχωράτε στο επόμενο.**

### 4. Άσκηση Νο 1 (System integration)

#### 4.1 Περιγραφή και στόχος της Άσκησης

Στόχος της άσκησης Νο 1 είναι η εξοικείωση με τη διαδικασία του System integration. Αυτό θα γίνει μέσα από την δημιουργία μιας αριθμομηχανής με γραφική διεπαφή που υποστηρίζει περιορισμένη λειτουργικότητα. Μπορεί να δεχθεί μια έκφραση που περιέχει τους τελεστές + και =, υπολογίζει την τιμή της και τη δίνει στον χρήστη.

Την έκδοση αυτή της RPNCalculator **θα αναπτύξετε χρησιμοποιώντας έτοιμα συνθετικά** που σας δίνονται με την μορφή βιβλιοθηκών.

#### 4.1.1 Η βιβλιοθήκη

Σας **δίνονται σε μορφή βιβλιοθηκών όλες οι κλάσεις** που απαιτούνται για την κατασκευή μιας έκδοσης που ικανοποιεί τον στόχο της άσκησης.

Εσείς **θα πρέπει να ασχοληθείτε μόνο με τη σύνθεση τους (integration)** ώστε να δημιουργηθεί η πρώτη δική σας RPN Calculator.

Σας δίδονται οι παρακάτω κλάσεις:

**CalculatorGui, Operand, Adder, Divider, Multiplier, Subtractor** and **ResultPresenter**.

Η κλάση Operand βρίσκεται στο πακέτο engine.

Οι κλάσεις Adder, Divider, Multiplier, Subtractor and ResultPresenter βρίσκονται στο πακέτο operators.

Η κλάση CalculatorGui βρίσκεται στο πακέτο gui.

Μια **τεκμηρίωση των κλάσεων** αυτών θα βρείτε στην ιστοσελίδα <http://seg.ceid.upatras.gr/Courses/OT/activities/activity3/doc/index.html>

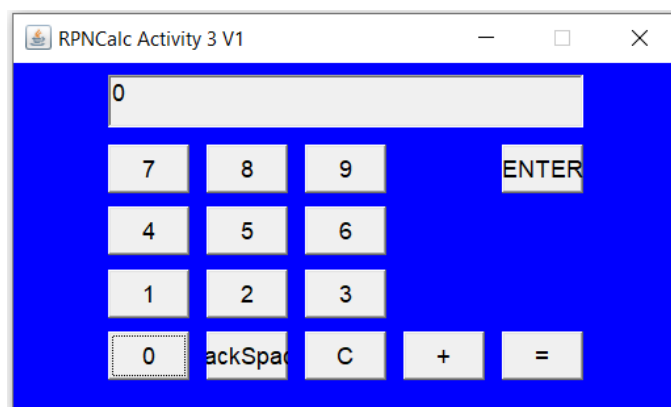
Τις κλάσεις (εκτός από την CalculatorGui) θα βρείτε στην **βιβλιοθήκη** η οποία σας δίνεται με την μορφή αρχείου τύπου jar (**Ex1-CalcComponents.jar**).

Την βιβλιοθήκη αυτή όπως και τις υπόλοιπες της EA θα την κατεβάσετε από τον παρακάτω σύνδεσμο

[https://github.com/ThramboulidisKleanthis/OOP\\_Course\\_Lab](https://github.com/ThramboulidisKleanthis/OOP_Course_Lab)

Την κλάση CalculatorGui θα χρησιμοποιήσετε από την βιβλιοθήκη **Ex1-CalcGui.jar**

Η CalculatorGui βρίσκεται στο πακέτο gui και ο δημιουργός της εμφανίζει στην βασική έξοδο μια γραφική διεπαφή όπως αυτή στο **Σχήμα 1**.



**Σχήμα 1.** Η γραφική διεπαφή που δημιουργεί ο constructor της CalculatorGui.

Μια σύντομη αναφορά στα αρχεία τύπου jar θα βρείτε στην [ενότητα 4.4](#).

## 4.2 Οδηγίες εκτέλεσης Άσκησης

### Δράση 1.1 – Ορισμός της κλάσης που αναπαριστά το πρόγραμμα σας

Με βάση τα παραπάνω και με τις μέχρι τώρα γνώσεις σας θα πρέπει να είστε σε θέση να ολοκληρώσετε μια πρώτη έκδοση της RPN Calculator.

Στην ενότητα 4.3 σας δίνονται αναλυτικές οδηγίες για την εκτέλεση της άσκησης. Στην ενότητα 4.4 σας δίνεται αναλυτική περιγραφή των συνιστωσών του προγράμματος. Ανατρέξτε σε αυτές μόνο μετά από τη δική σας προσπάθεια. Αξιοποιήστε επιλεκτικά όσο λιγότερες από αυτές μπορείτε.

Με την ολοκλήρωση της Δράσης 1 προχωρήστε στην 2 και την 3 για να ολοκληρώσετε την Άσκηση No 1.

### Δράση 1.2 – Έλεγχος λειτουργίας

Ελέγξτε το πρόγραμμα σας για την ορθή λειτουργία του. Θα πρέπει να εμφανίζει την γραφική διεπαφή η οποία θα σας επιτρέψει να εισάγεται τελεστέους και να κάνετε την πράξη της πρόσθεσης (+) και της παρουσίασης του αποτελέσματος (=). Πατήστε το ENTER για να σηματοδοτήσετε το τέλος του τελεστέου.

Παρατηρήστε ότι το πρόγραμμα τυπώνει την τρέχουσα τιμή του τελεστέου στην βασική έξοδο και όχι στο display της γραφικής διεπαφής. Στη φάση αυτή δεν μπορείτε να επέμβετε σε αυτό. Θα το αντιμετωπίσετε στα πλαίσια της άσκησης.

### Δράση 1.3 – Επέκταση λειτουργικότητας

Τροποποιήστε το πρόγραμμα που αναπτύξατε παραπάνω ώστε να τυπώνει την κατάσταση της στοίβας κάθε φορά που ολοκληρώνεται ένας τελεστής και μπαίνει στην στοίβα. Αυτό χωρίς να επέμβετε στην κλάση Operand που σας δίνεται στο jar.

## 4.3 Αναλυτικές Οδηγίες Εκτέλεσης της άσκησης No 1

Σας δίνονται οδηγίες για:

1. [χρήση του Eclipse](#) για την ανάπτυξη του αντίστοιχου project.
2. αξιοποίησης βιβλιοθήκης με την μορφή αρχείου τύπου jar.
3. χρήση των κλάσεων της βιβλιοθήκης.

### 4.3.1 Δημιουργία project στο Eclipse

Αν επιλέξατε ως IDE το Eclipse εκτελέστε τις παρακάτω ενέργειες για την δημιουργία του Project.

#### **E1.1.** Δημιουργήστε Project

Επιλέξτε

**File ->New->Java Project.**

Δώστε όνομα στο project και επιλέξτε Finish και στην συνέχεια στο παράθυρο Create module-info.java επιλέξτε Don't Create για **να μην δημιουργηθεί module**. Σε νεότερη έκδοση του Eclipse αν η δημιουργία module είναι προεπιλεγμένη απενεργοποιήστε την.

Δημιουργήστε κάτω από το ευρετήριο src του project σας ένα package με όνομα tester ή ότι άλλο όνομα θέλετε.

#### **E1.2.** Δημιουργήστε την κλάση που αναπαριστά το πρόγραμμα σας.

Στον package explorer επιλέξτε το project και στη συνέχεια το πακέτο που δημιουργήσατε. Δεξί κλικ και **New-> Class**.

Δώστε όνομα Calc (ή RPNCalculator ή όποιο άλλο θέλετε) και ζητήστε τη δημιουργία της main() (Σχήμα 2). Πατήστε Finish.

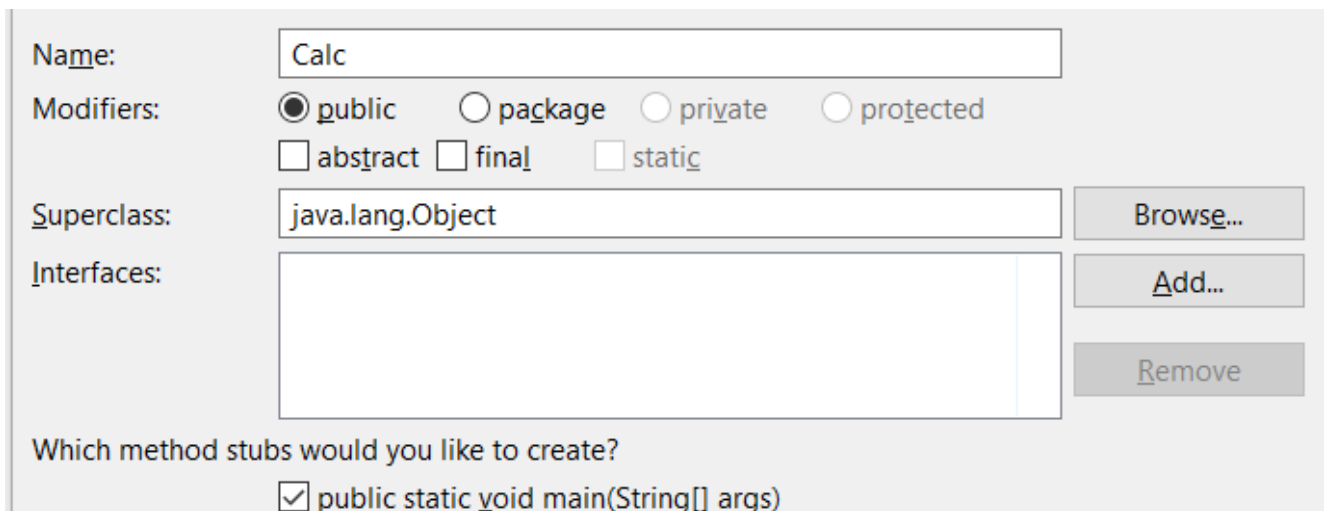
#### **E1.3.** Ορίστε τη μέθοδο main.

Πρώτα από όλα δοκιμάστε την διαδικασία μεταγλώττισης και εκτέλεσης με μία απλή System.out.println. Προχωρήστε στη συνέχεια στην συγγραφή της main.

Η main θα πρέπει να δημιουργήσει ένα στιγμιότυπο της γραφικής διεπαφής. Αυτό θα το κάνει καλώντας κατάλληλα τον δημιουργό της CalculatorGui.

Για να αξιοποιήσετε κλάσεις της βιβλιοθήκης θα πρέπει να έχετε ενημερώσει κατάλληλα το IDE σας.

Για την αξιοποίηση κλάσης των βιβλιοθηκών θα πρέπει να περιλάβετε στον πηγαίο κώδικα την αντίστοιχη πρόταση **import**. Το Eclipse σας δίνει βοήθεια προς την κατεύθυνση αυτή.



**Σχήμα 2** Δημιουργία κλάσης Calc στο Eclipse.

### 4.3.2 Αξιοποίηση βιβλιοθηκών τύπου Jar

Στην ενότητα αυτή δίνονται οδηγίες για αξιοποίηση βιβλιοθηκών τύπου jar. Περισσότερες πληροφορίες για αρχεία τύπου jar θα βρείτε στη ενότητα [Αρχεία τύπου jar \(Java Archive files\)](#)

Για να αξιοποιήσετε βιβλιοθήκες τύπου .jar εκτελέστε τις παρακάτω ενέργειες:

**E1.4.** – Κατεβάστε τις βιβλιοθήκες

Αντιγράψτε τις βιβλιοθήκες στο ευρετήριο του project που δημιουργήσατε.

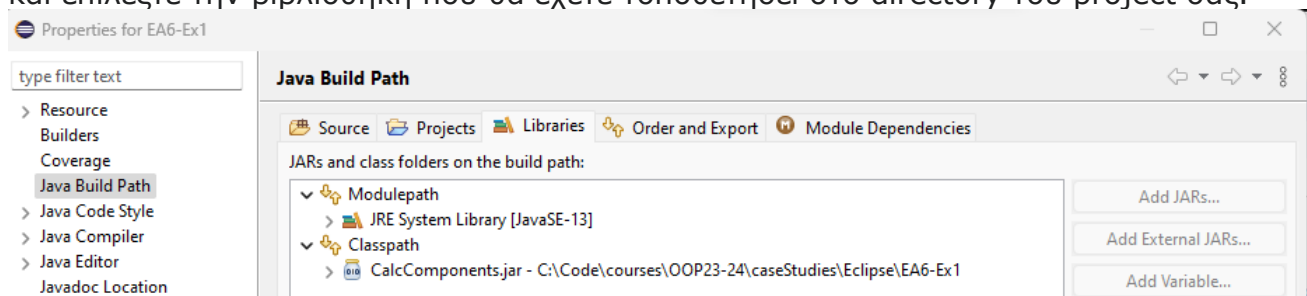
**E1.5.** - Ενημερώστε το IDE

#### A) Για Eclipse

Ενημερώστε το Eclipse (ή όποιο IDE χρησιμοποιείτε) για την θέση στην οποία θα βρει τις βιβλιοθήκες που χρησιμοποιεί το project σας. Επιλέξτε στον Package Explorer το project και δεξί κλικ και επιλέξτε

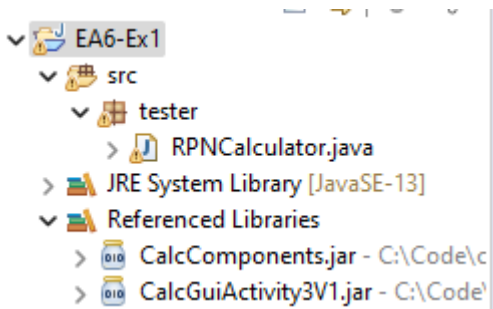
#### ***Project-Properties-Java Build Path-Libraries-ClassPath-Add external JARs***

και επιλέξτε την βιβλιοθήκη που θα έχετε τοποθετήσει στο directory του project σας.



Επιλέξτε Apply and Close.

Στον project Explorer θα τις δείτε κάτω από το **Referenced Libraries** όπως φαίνεται παρακάτω.



## B) Για BlueJ

Για ενημέρωση του BlueJ για την βιβλιοθήκη κλάσεων επιλέξτε

**Tools-Preferences-Libraries-Add** και επιλέξτε την βιβλιοθήκη `CalcGuiActivity3V1.jar`

Αν χρησιμοποιείτε άλλο IDE αναζητήστε στο διαδίκτυο πως το κάνετε ενημέρωση για χρήση αρχείων jar.

## 4.4 Περιγραφή των συνιστωσών του προγράμματος

Μια προσεκτική μελέτη του προβλήματος με στόχο την ικανοποίηση των επιδιώξεων της EA οδήγησε στον ορισμό της δομής της RPN Calculator ως συνάθροισης από τα παρακάτω αντικείμενα.

### 4.4.1 CalculatorGui

Το βασικότερο, για τον χρήστη, αντικείμενο της εφαρμογής είναι αυτό που του επιτρέπει να χρησιμοποιήσει την υπηρεσία που περιγράφει το πρόγραμμα μας και θα προσφέρει η μηχανή που θα εκτελεί το πρόγραμμα μας.

Επιλέξαμε στη φάση αυτή η επικοινωνία ανθρώπου με την μηχανή να υλοποιείται με χρήση γραφικής διεπαφής (**Σχήμα 1**).

Προσέξτε όμως, η γραφική διεπαφή δεν εκτελεί καμία ουσιαστική λειτουργία εκτός από το να επιτρέπει στον χρήστη να αλληλοεπιδρά με τα αντικείμενα που υλοποιούν τη λειτουργικότητα της αριθμομηχανής μας, την RPN Calculator engine. Απλά θα στέλνει

μηνύματα στα αντικείμενα που έχουν επιφορτιστεί με τις κατάλληλες αρμοδιότητες ώστε η συνεργασία τους να μας δώσει την υπηρεσία που θέλουμε σε επίπεδο συστήματος.

Η κλάση `CalculatorGui` θα σας δώσει και τη δυνατότητα να κατανοήσετε τις αρμοδιότητες που έχουν ανατεθεί στα αντικείμενα που απαρτίζουν την μηχανή (engine) της αριθμομηχανής μας.

#### 4.4.2 Operand

Βασική κλάση που αναπαριστά δομή και συμπεριφορά του τελεστέου ο οποίος είναι βασικό συνθετικό της εισόδου που δέχεται το σύστημα.

Δίνει τη δυνατότητα διαμόρφωσης του τελεστέου από τη γραφική διεπαφή οπότε πρέπει να υποστηρίζει την προσθήκη ψηφίου στην τιμή του τελεστέου (θα ονομάσουμε την αντίστοιχη μέθοδο `addDigit`), τη διαγραφή του τελευταίου εισαχθέντος ψηφίου (θα ονομάσουμε την αντίστοιχη μέθοδο `deleteLastDigit`), την ολοκλήρωση του τελεστέου και την αποθήκευση του στην στοίβα (θα ονομάσουμε την αντίστοιχη μέθοδο `complete`) κλπ.

#### 4.4.3 Adder, Divider, Multiplier, και Subtractor

Η κλάση `Adder` αναπαριστά τον αθροιστή που δουλεύει πάνω σε στοίβα. Την κλάση αυτή έχουμε ήδη μελετήσει αναλυτικά σε προηγούμενη ΕΑ.

Αντίστοιχες είναι και οι άλλες κλάσεις που αναπαριστούν τους υπόλοιπους τελεστές που υποστηρίζει η έκδοση αυτή.

#### 4.4.4 ResultPresenter

Η κλάση `ResultPresenter` αναπαριστά τον τελεστή = και η λειτουργικότητα της είναι η εμφάνιση στη οθόνη του τελευταίου στοιχείου της στοίβας.

#### 4.4.5 RPNCalculator

Είναι η κλάση που θα αναπαριστά το πρόγραμμα και για τον λόγο αυτό θα την ονομάσουμε `RPNCalculator`. Αυτή θα περιέχει την `main()` η οποία θα πρέπει να εμφανίσει τη γραφική διεπαφή.

Προσέξτε πως για την κλήση του δημιουργού της κλάσης της γραφικής διεπαφής θα πρέπει να έχετε πρώτα δημιουργήσει στιγμιότυπα των οποίων αναφορές πρέπει να περάσετε στον δημιουργό κατά την κλήση του.



**Έχει δηλαδή έχει την ευθύνη να δημιουργήσει τα αντικείμενα που απαρτίζουν το σύστημα και να αποκαταστήσει τις μεταξύ τους εξαρτήσεις.**

## 4.5 Αρχεία τύπου jar (Java Archive files)

Τα αρχεία τύπου jar (Java Archive files) αποτελούν την καθιερωμένη μέθοδο ενσωμάτωσης πολλών class αρχείων σε ένα. Το αρχείο αυτό μπορεί να χρησιμοποιηθεί ως βιβλιοθήκη κλάσεων για την ανάπτυξη Java εφαρμογών.

Για περισσότερες πληροφορίες σχετικά με τα αρχεία τύπου jar και τον τρόπο δημιουργίας και χρήσης τους μπορείτε να ανατρέξετε στη διεύθυνση:

<https://docs.oracle.com/javase/tutorial/deployment/jar/basicindex.html>

### 4.5.1 Χρησιμοποιώντας αρχεία jar με το SDK της Sun.

Για να αξιοποιήσουμε σε ένα πρόγραμμα κλάσεις που περιέχονται σε μια βιβλιοθήκη με όνομα "myLibrary.jar" ενημερώνουμε τον Java compiler για το όνομα και το ευρετήριο στο οποίο βρίσκεται η βιβλιοθήκη.

Για παράδειγμα, αν υποθέσουμε ότι η κλάση που αναπαριστά το πρόγραμμα είναι η MyProgram και ότι η βιβλιοθήκη myLibrary.jar βρίσκεται στον κατάλογο c:\libraries καλούμε τον java compiler όπως παρακάτω:

```
javac -classpath ".;c:\libraries\myLibrary.jar" *.java
```

και εκτελούμε το πρόγραμμα μας καλώντας τον java interpreter όπως παρακάτω:

```
java -classpath ".;c:\libraries\myLibrary.jar" MyProgram
```

Για πληροφορίες σχετικά με τη μεταβλητή περιβάλλοντος "classpath" μπορείτε να ανατρέξετε στην ιστοσελίδα :

<http://java.sun.com/j2se/1.3/docs/tooldocs/win32/classpath.html>

## 5. Άσκηση No 2 - Ανάπτυξη συνθετικών της RPN Calculator

### 5.1 Περιγραφή και στόχος της Άσκησης

Στην άσκηση αυτή θα αναπτύξετε μια RPN Calculator με δικά σας συνθετικά. Δεν θα χρησιμοποιήσετε δηλαδή τα έτοιμα συνθετικά που σας δόθηκαν στην Άσκηση1 αλλά θα αναπτύξετε τα δικά μας. Εξαίρεση αποτελεί η γραφική διεπαφή την οποία θα αναπτύξετε στην επόμενη άσκηση (Άσκηση 3).

### 5.2 Οδηγίες εκτέλεσης Άσκησης (Διαδικασία Ανάπτυξης)

Θα εφαρμόσετε την incremental development τεχνική για την ανάπτυξη μιας λειτουργούσας έκδοσης της RPN Calculator.

Περιγράφονται στη συνέχεια ένα σύνολο από επαναλήψεις (iterations) που οδηγούν σε μια λειτουργούσα αριθμομηχανή η οποία θα χρησιμοποιεί μόνο την γραφική διεπαφή ως έτοιμη κλάση.

Ανάλογα με το επίπεδο σας μπορείτε να επιλέξετε να συμπύξετε 2 ή περισσότερα iterations σε ένα.

Στην επόμενη ενότητα (5.2) σας δίνονται οδηγίες για κάθε iteration. Ανατρέξτε σε αυτές μόνο μετά από την δική σας προσπάθεια. Αξιοποιήστε επιλεκτικά όσες λιγότερες από αυτές μπορείτε.

#### 5.2.1 Iteration No 1 – Εμφάνιση γραφικής διεπαφής

Το iteration αυτό έχει στόχο την ανάπτυξη της πρώτης έκδοσης της RpnCalculator η οποία απλά θα εμφανίζει την γραφική διεπαφή.

Ως γραφική διεπαφή θα χρησιμοποιήσετε αυτή που σας δίνεται στο αρχείο **Ex2It1- CalcGui.jar** (github)

#### Δράση 2.1

Δημιουργήστε ένα project με όνομα **LabEx6-It1**. Δουλέψτε στο default package.

Ο δημιουργός της `CalculatorGui` δέχεται ένα όρισμα τύπου Operand.

Δώστε τον πηγαίο κώδικα δουλεύοντας με μία dummy Operand. Στην οθόνη θα πρέπει να δείτε μια γραφική διεπαφή όπως αυτή στο Σχήμα 3.



**Σχήμα 3** Η γραφική διεπαφή του 1ου iteration.

Πατήστε τα πλήκτρα των ψηφίων, το Backspace και το ENTER. Εξηγήστε την συμπεριφορά του προγράμματος.

### 5.2.2 Iteration No 2 – Ορισμός της κλάσης Operand

Στο iteration αυτό θα δημιουργήσουμε μια 2<sup>η</sup> έκδοση της RpnCalculator. Η έκδοση αυτή θα μας επιτρέψει να διαμορφώνουμε από την γραφική διεπαφή έναν τελεστέο (Operand) και να τον βάζουμε στη στοίβα. Η απαίτηση του να διαμορφώνουμε τον τελεστέο από τη γραφική διεπαφή μας οδηγεί στον ορισμό της κλάσης Operand. Στην Άσκηση 1 την είχαμε έτοιμη. Εδώ θα πρέπει να την ορίσετε.

#### **Δράση 2.2**

Δημιουργήστε ένα αντίγραφο του project του 1<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό. Θα πρέπει να ορίσετε το σώμα της Operand. Εξετάστε δύο τουλάχιστον υλοποιήσεις της Operand.

#### **Δράση 2.3**

Τροποποιήστε το πρόγραμμά σας ώστε να ελέγξετε αν ο operand που ορίσατε από την γραφική διεπαφή μπήκε στη στοίβα.

Στην επόμενη έκδοση θα προσθέσουμε στο πρόγραμμα και τους τελεστές για να κάνουμε πράξεις.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο Ch6\_LegoApproachPartC.

### 5.2.3 Iteration No 3 – Ορισμός λοιπών κλάσεων

Στην έκδοση αυτή (3<sup>η</sup> έκδοση) θα προσθέσουμε τη λειτουργικότητα των πράξεων και της εμφάνισης του αποτελέσματος. Θα ορίσετε δηλαδή τις δικές σας κλάσεις *Adder*, *Subtracter*, *Multiplier*, κλπ.

#### Δράση 2.4

Δημιουργήστε ένα αντίγραφο του project του 2<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό. Καθώς όμως η γραφική διεπαφή της έκδοσης 3 δεν έχει τελεστές θα αντικαταστήσετε την γραφική διεπαφή με αυτή που περιέχεται στο αρχείο **Ex2It3-CalcGui.jar**

Η κλάση έχει τον παρακάτω δημιουργό

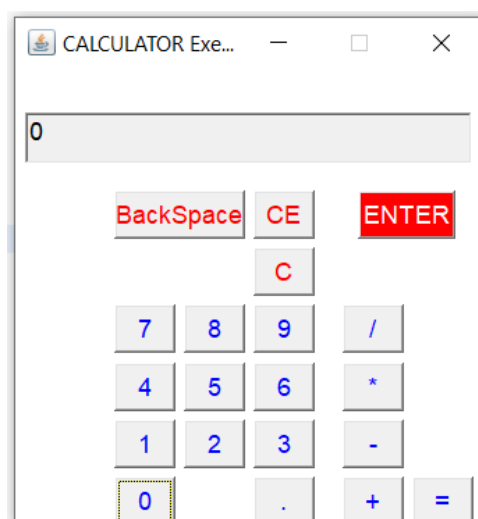
```
CalculatorGui(Operand op, Adder add, Subtracter sub, Multiplier mul, Divider div, ResultPresenter rp)
```

Ο δημιουργός αυτός εμφανίζει στην οθόνη μία γραφική διεπαφή όπως αυτή στο **Σχήμα 4**.

#### Δράση 2.5

Ορίστε κατάλληλα τις δικές σας κλάσεις για τους τελεστές και προχωρήστε στην σύνθεση της εφαρμογής σας. Την *Adder* την έχετε έτοιμη. Ανάλογα θα ορίσετε και τις υπόλοιπες.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο *Ch6\_LegoApproachPartC*.



**Σχήμα 4.** Η γραφική διεπαφή του 4ου iteration.

## 5.2.4 Iteration No 4 – Αξιοποίηση της Κληρονομικότητας (inheritance)

Στην έκδοση αυτή δεν θα προσθέσετε επιπλέον λειτουργικότητα. Θα τροποποιήσετε τον κώδικα ώστε να αξιοποιήσετε τον μηχανισμό της κληρονομικότητας.

### Δράση 2.6

Αναγνωρίστε σχέση γενίκευσης/εξειδίκευσης στο πρόβλημα σας (ή στον κώδικα σας) και δώστε ένα διάγραμμα κλάσεων που καταγράφει την σχέση αυτή.

### Δράση 2.7

Δημιουργήστε ένα αντίγραφο του project του 3<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό. Αξιοποιήστε τον μηχανισμό της κληρονομικότητας (inheritance) για να υλοποιήσετε την σχέση γενίκευσης/εξειδίκευσης που καταγράψατε στο διάγραμμα κλάσεων.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις. Περιγράφεται στις διαφάνειες με τίτλο Ch6\_LegoApproachPartC.

## 5.2.5 Iteration No 5 – Αξιοποίηση του Interface

Στην έκδοση αυτή δεν θα προσθέσουμε επιπλέον λειτουργικότητα. Θα τροποποιήσουμε τον κώδικα της Operand και της CalculatorGui ώστε να αξιοποιούν την κατασκευή του interface.

Το iteration αυτό θα το δουλέψετε όταν έχετε ένα πρώτο δικό σας ορισμό της κλάσης CalculatorGui. Αυτό θα το κάνετε στην επόμενη Άσκηση ή μετά το iteration No 6.

### Δράση 2.8

Ορίστε το interface `OperandIf` έτσι ώστε η κλάση `Operand` να ορίζεται να κάνει implements το interface αυτό.

Την όλη διαδικασία συζητήσαμε στις διαλέξεις.

## 5.2.6 Ερώτηση

Ποια από τις 3 εκδόσεις (iteration 3,4, 5) θα προτιμήσετε και γιατί; Αιτιολογήστε.

## 5.2.7 Iteration No 6 – Επέκταση λειτουργικότητας και βελτιώσεις πηγαίου κώδικα

Ο στόχος αυτού του iteration είναι να επεκτείνετε λειτουργικότητα και να βελτιώσετε πηγαίο κώδικα του οποίου τη σημασία δεν κατανοείτε στην φάση αυτή πλήρως. Ο πηγαίος κώδικας που σας δίνεται αφορά τη γραφική διεπαφή της αριθμομηχανής και περιλαμβάνει λάθη και κακές επιλογές που συνήθως υιοθετούν αρχάριοι προγραμματιστές. Προσπαθήστε να εκτελέσετε το iteration αυτό με τα όσα έχετε διδαχθεί μέχρι σήμερα. Τα περί γραφικών διεπαφών θα δούμε αναλυτικά σε επόμενες διαλέξεις.

### Δράση 2.9

Δημιουργήστε ένα αντίγραφο του iteration 3 και δουλέψτε πάνω σε αυτό.

Κάντε τις απαραίτητες τροποποιήσεις στο project σας ώστε να χρησιμοποιεί για γραφική διεπαφή αντί για την κλάση του αρχείου Ex2It3-CalcGui.jar την κλάση CalculatorGui η οποία σας δίνεται σε μορφή πηγαίου κώδικα (αρχείο **Ex2It6-CalculatorGui.zip**).

Την κλάση CalculatorGui συνοδεύει και ένα σύνολο από άλλες βοηθητικές κλάσεις οι οποίες έχουν τοποθετηθεί στο αρχείο της κλάσης CalculatorGui.

Ως πρώτο βήμα διορθώστε λάθη που αναγνωρίζει το IDE. Αξιοποιήστε την βοήθεια που σας προσφέρει το IDE σας προς την κατεύθυνση αυτή.

Εκτελέστε το πρόγραμμα σας για να ελέγξετε την λειτουργικότητα του.

### Δράση 2.10 Επέκταση λειτουργικότητας

Τροποποιήστε την γραφική διεπαφή ώστε να εκτελεί τις πράξεις του πολλαπλασιασμού και της διαίρεσης. Στην φάση αυτή δεν είναι απαραίτητο να κατανοήσετε τον κώδικα.

Προσθέστε την νέα λειτουργικότητα αξιοποιώντας τον ήδη υπάρχοντα κώδικα.

Ελέγξτε τη συμπεριφορά της αριθμομηχανής στα πλήκτρα C και CE.

### Δράση 2.11 Κακές πρακτικές

Εντοπίστε κακές πρακτικές και τροποποιήστε τον κώδικα σας με στόχο την καλύτερη εφαρμογή του αντικειμενοστρεφούς παραδείγματος.

## 5.3 Αναλυτικές Οδηγίες Εκτέλεσης της άσκησης No 2

### 5.3.1 Iteration No 1 – Εμφάνιση γραφικής διεπαφής

Δημιουργήστε ένα project με όνομα **LabEx6-It1**.

Κάντε δεξί κλικ στον φάκελο src του project σας και επιλέξτε New->Class. Δώστε όνομα στην κλάση σας και επιλέξτε κάτω από το πλαίσιο Interfaces το κουτάκι

```
public static void main(String[] args)
```

Πατήστε Finish.

Κατεβάστε το αρχείο της γραφικής διεπαφής και ενημερώστε για αυτό το Eclipse.

Ενημερώστε την main ώστε να δημιουργεί ένα στιγμιότυπο της γραφικής διεπαφής. Ο δημιουργός της CalculatorGui είναι ο παρακάτω

```
CalculatorGui(Operand op);
```

Μπορείτε να τον χρησιμοποιήσετε με μια dummy Operand, δηλαδή μια Operand χωρίς σώμα. Θα ορίσετε το σώμα της Operand στο επόμενο iteration.

Εκτελέστε το πρόγραμμα σας. Παρατηρήστε και εξηγήστε την συμπεριφορά του.

### 5.3.2 Iteration No 2 – Ορισμός της κλάσης Operand

Για την διαμόρφωση του operand θα πρέπει να ορίσετε την μέθοδο **addDigit()**.

Για την τοποθέτηση του τελεστέου στην στοιβα (πλήκτρο ENTER) πρέπει να ορίσετε τη μέθοδο **complete()**.

Εξετάστε δύο τουλάχιστον υλοποιήσεις της Operand. Μία με μεταβλητή πρωτογενούς τύπου int και μία με αλφαριθμητικό.

### 5.3.3 Iteration No 6 – Επέκταση λειτουργικότητας και βελτιώσεις πηγαίου κώδικα

Διαγράψτε την βιβλιοθήκη Ex2It3-CalcGui.jar από το ClassPath (Properties) του project σας και προσθέστε την κλάση CalculatorGui στο default package.

Τα λάθη που σας ζητείται να διορθώσετε έχουν να κάνουν μόνο με import των κλάσεων του javaAPI. Ορισμένα από αυτά (αυτά στα οποία το Eclipse δεν προτείνει το κατάλληλο import) διορθώνονται κινούμενοι από κάτω προς τα πάνω.

Το πρώτο από αυτά είναι στην πρόταση

```
button0.addActionListener(new Button0Handler());
```

Για τη διόρθωση του λάθους αυτού διορθώστε πρώτα το λάθος με τα κατάλληλα import στον ορισμό της κλάσης Button0Handler

```
class Button0Handler implements ActionListener{
```

Αυτά είναι ένα για την κλάση ActionListener και ένα για την κλάση ActionEvent.

Ένα αντίγραφο της CalculatorGui με τις διορθώσεις μπορείτε να βρείτε στο αρχείο **Ex2It6-**

**CalculatorGui-ErrorsCorrected.zip**



## 6. Άσκηση No 3 - Ανάπτυξη γραφικής διεπαφής της RPN Calculator (Gui και Event Handling)

### 6.1 Περιγραφή και στόχος της Άσκησης

Στην άσκηση αυτή θα σχεδιάσετε και θα υλοποιήσετε την δική σας γραφική διεπαφή για την RPN Calculator σας η οποία θα υλοποιεί την λειτουργικότητα που εσείς θα επιλέξετε.

Η άσκηση αυτή έχει επιπλέον στόχο την εξοικείωση σας με:

1. με τις γραφικές διεπαφές της Java (e.g., Abstract Window Toolkit ή/και swing),
2. με τον μηχανισμό διαχείρισης συμβάντων (event handling) της Java,
3. την κατασκευή του Interface, του μηχανισμού του package και των nested και anonymous κλάσεων,
4. Τον Χειρισμό εξαιρέσεων (Exception Handling),
5. τους μηχανισμούς οργάνωσης του κώδικα (Packages, Access level modifiers, Mutators and accessors, Abstract Methods and Classes, initializers) για την βελτίωση της ποιότητας του,
6. την αποφυγή χρήσης της πρότασης switch.

Σας συνιστούμε να εφαρμόσετε την incremental development τεχνική για την ανάπτυξη της δικής σας γραφικής διεπαφής η οποία θα αντικαταστήσει την γραφική διεπαφή που σας δόθηκε έτοιμη.

Σχεδιάστε πρώτα την δική σας γραφική διεπαφή. Δημιουργήστε το διάγραμμα κλάσεων της και στη συνέχεια προχωρήστε στην συγγραφή του κώδικα σας.

### 6.2 Οδηγίες εκτέλεσης Άσκησης

Καθώς ο στόχος της άσκησης αυτής είναι η ανάπτυξη της δικής γραφικής διεπαφής θα ξεκινήσετε πρώτα από τον σχεδιασμό της και στη συνέχεια θα περάσετε στην συγγραφή του κώδικα που θα την υλοποιεί.

#### **Δράση 3.1 – Σχεδιασμός γραφικής διεπαφής**

Σχεδιάστε με χαρτί και μολύβι την γραφική διεπαφή για την αριθμομηχανή σας.

#### **Δράση 3.2 – Διάγραμμα Κλάσεων Γραφικής Διεπαφής**

1. Αναγνωρίστε τα αντικείμενα που απαρτίζουν την γραφική σας διεπαφή και δημιουργήστε ένα διάγραμμα κλάσεων για την γραφική σας διεπαφή.
2. Περιλάβετε σε αυτό και τις κλάσεις της βασικής βιβλιοθήκης τις οποίες οι δικές σας κλάσεις άμεσα κληρονομούν (όπως εσείς τις έχετε ορίσει).
3. Περιλάβετε σε αυτό και τα interfaces της βασικής βιβλιοθήκης τα οποία οι δικές σας κλάσεις υλοποιούν.

Αν σας δυσκολεύει η δημιουργία του διαγράμματος κλάσεων μπορείτε να το διαμορφώσετε παράλληλα με την συγγραφή του κώδικα σας.

### **Δράση 3.3 – Συγγραφή κώδικα Γραφικής Διεπαφής**

Η δράση αυτή είναι πολύ σύνθετη και περιγράφεται αναλυτικά στην επόμενη ενότητα.

## **6.3 Συγγραφή πηγαίου κώδικα**

### **6.3.1 Γενικές οδηγίες**

Αναπτύξτε τον κώδικα που θα υλοποιεί την γραφική διεπαφή που σχεδιάσατε.

Κάντε προσπάθεια να αξιοποιήσετε όσο το δυνατόν περισσότερες από τις κατασκευές της γλώσσας για να αυξήσετε το reusability και το modularity του κώδικα σας.

Στη συνέχεια περιγράφονται ένα σύνολο από iterations που θα σας βοηθήσουν να φτάσετε βήμα προς βήμα στον στόχο σας, την υλοποίηση της δικής γραφικής διεπαφής. Ανάλογα με το επίπεδο σας μπορείτε να επιλέξετε να συμπτύξετε 2 ή περισσότερα iterations σε ένα.

Τα iterations 1-3 έχουν στόχο μια πρώτη πολύ βασική έκδοση της CalculatorGui η οποία δημιουργεί ένα παράθυρο και βάζει μέσα σε αυτό ένα button για το οποίο προσδιορίζει τη συμπεριφορά του στο πάτημα του. Επιπλέον, δημιουργεί και το display της γραφικής διεπαφής.

Τα iterations 4-5 έχουν στόχο μια πιο βελτιωμένη έκδοση της γραφικής διεπαφής που θα σας επιτρέψει να δημιουργήσετε μια έκδοση της αριθμομηχανής σας που θα υπολογίζει την τιμή μιας απλής έκφρασης, όπως π.χ. της  $100 \cdot 10 + =$ .

Το iteration 6 έχει στόχο την δημιουργία διεπαφής μιας πλήρως λειτουργικής αριθμομηχανής.

Για κάθε σας δίνονται και αναλυτικές οδηγίες στην ενότητα 6.4. Ορισμένες από αυτές επιβάλλεται να τις μελετήσετε.

### 6.3.2 Iteration No 1 – Δημιουργία Frame

Ορίστε την κλάση CalculatorGui να κληρονομεί τη Frame (πακέτο java.awt) ή εναλλακτικά την JFrame αν θέλετε να δουλέψετε με το πακέτο swing.

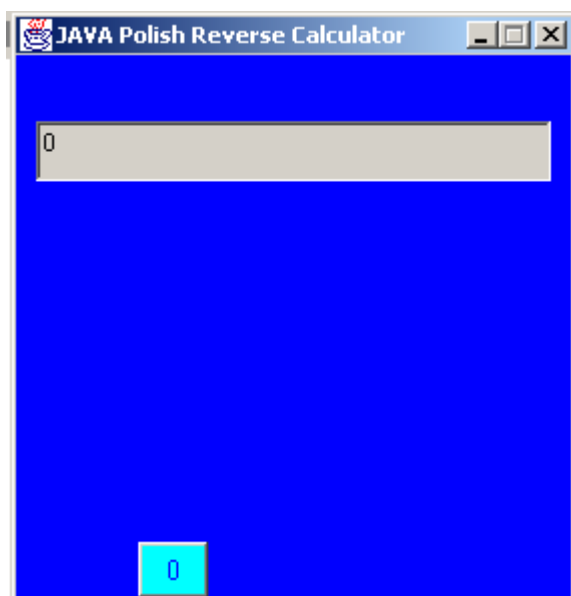
**E3.1.** Ορίστε στον δημιουργό της CalculatorGui τα χαρακτηριστικά του frame που δημιουργεί ο δημιουργός της Frame.

**E3.2.** Καλέστε το δημιουργό της CalculatorGui για να δείτε το αποτέλεσμα.

Αναλυτικές οδηγίες μπορείτε να βρείτε στην ενότητα 6.4

### 6.3.3 Iteration No 2 – Προσθήκη Button και ορισμός συμπεριφοράς

Δημιουργήστε ένα αντίγραφο του project του 1<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό. Στο 2<sup>ο</sup> αυτό iteration θα προσθέσετε ένα button στη γραφική διεπαφή και θα του προσδιορίσετε συμπεριφορά ως απόκριση στο συμβάν button pressed.

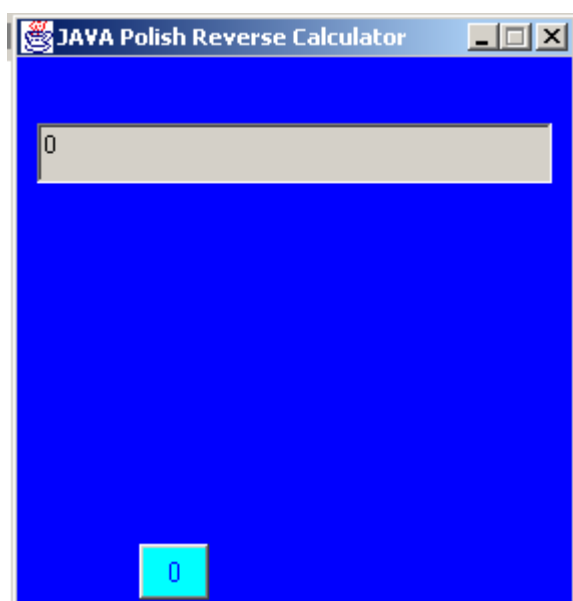


Ανεξάρτητα από την δική σας προσπάθεια θα πρέπει να ανατρέξετε και στις αναλυτικές οδηγίες για το iteration αυτό.

Στο επόμενο iteration θα προσθέσουμε στη γραφική διεπαφή το display.

### 6.3.4 Iteration No 3 – Προσθήκη display

Στο 3ο αυτό iteration θα προσθέσουμε ένα display στη γραφική διεπαφή και θα του προσδιορίσουμε την συμπεριφορά του ως απόκριση στο συμβάν button pressed να είναι η εμφάνιση του μηνύματος "Button 0 pressed" στο display.



### 6.3.5 Iteration No 4 – Προσθήκη της Operand

Δημιουργήστε ένα αντίγραφο του project του 3<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό.

Στο 4<sup>ο</sup> αυτό iteration θα προσθέσουμε στο project μας την κλάση Operand και θα τροποποιήσουμε την συμπεριφορά του button0 ώστε να στέλνει με το πάτημα του το μήνυμα addDigit σε ένα στιγμιότυπο της Operand. Έτσι πατώντας τα πλήκτρα των ψηφίων (τα οποία θα βάλετε στη συνέχεια στη γραφική σας διεπαφή) θα ορίζετε την τιμή του τελεστή.

### 6.3.6 Iteration No 5 – Προσθήκη των Adder και ResultPresenter

Δημιουργήστε ένα αντίγραφο του project του 4ου iteration και δουλέψτε πάνω σε αυτό.

Στο 5ο αυτό iteration θα προσθέσουμε στο project μας τις κλάσεις Adder και ResultPresenter. Τις έχετε ήδη έτοιμες από προηγούμενη Δραστηριότητα.

Αν εκτελέσατε το iteration αυτό, η ολοκλήρωση της RPNCalculator με τη δική σας γραφική διεπαφή είναι μηχανική διαδικασία. Η RPNCalculator σας θέλει πλέον μόνο λίγο χρόνο για να ολοκληρωθεί.

### **6.3.7 Iteration No 6 – Γραφική διεπαφή μιας πλήρως λειτουργικής αριθμομηχανής**

Δημιουργήστε ένα αντίγραφο του project του 5<sup>ου</sup> ή 6<sup>ου</sup> iteration και δουλέψτε πάνω σε αυτό.

**E3.13.** Διαμορφώστε τη γραφική διεπαφή έτσι ώστε να σας δίνει τη δυνατότητα να αξιοποιήσετε όλη την λειτουργικότητα που έχετε προσθέσει στην αριθμομηχανή σας. Προσθέστε την επιπλέον λειτουργικότητα που απαιτεί η γραφική σας διεπαφή.

## **6.4 Αναλυτικές Οδηγίες Εκτέλεσης της άσκησης No 3**

### **6.4.1 Iteration No 1 – Δημιουργία Frame**

Το

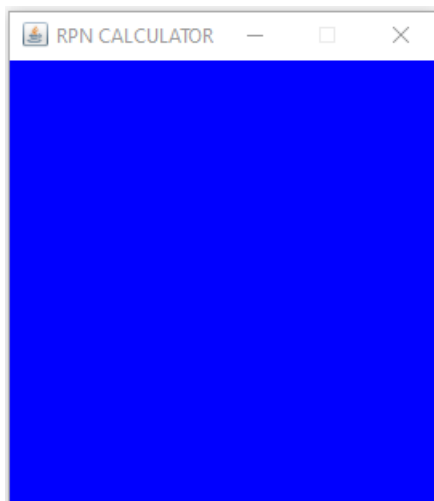
Σχήμα 6.1 δίνει έναν ορισμό της για το iteration1 και το Σχήμα 6.2 δίνει το αποτέλεσμα της κλήσης του δημιουργού της CalculatorGui όπως προκύπτει από την εκτέλεση του προγράμματος RPNCalculator που δίνεται στο Σχήμα 6.3. Το Σχήμα 6.4 δίνει τον ορισμό της κλάσης CalculatorGui για το iteration1 με χρήση της βιβλιοθήκης swing.

```

RpnCalculator.java  RpnCalculatorGui.java ×
1 import java.awt.*;
2 import java.awt.event.*;
3
4 public class RpnCalculatorGui extends Frame{
5
6     public RpnCalculatorGui(String title)
7     {
8         super(title);
9         this.setLayout(null);
10        this.setFont(new Font("TimesRoman", Font.PLAIN, 14));
11        this.setBackground(Color.blue);
12
13        this.setSize(283,320);
14        this.setLocation(40,80);
15        this.setVisible(true);
16        this.setResizable(false);
17        this.addWindowListener(new CloseWindowAndExit());
18    }
19 }
20 class CloseWindowAndExit extends WindowAdapter {
21     public void windowClosing(WindowEvent closeWindowAndExit) {
22         System.exit(0);
23     }
24 }

```

Σχήμα 6.1 Ορισμός της κλάσης CalculatorGui για το iteration1



Σχήμα 6.2 Το αποτέλεσμα της κλήσης του δημιουργού της CalculatorGui.

```

RpnCalculator.java × RpnCalculatorGui.java
1 public class RpnCalculator {
2     static RpnCalculatorGui cg;
3
4     public static void main(String[] args) {
5         cg = new RpnCalculatorGui("RPN CALCULATOR");
6     }
7 }

```

Σχήμα 6.3 Ορισμός της κλάσης RPNCalculator.

```

RpnCalculator.java CalculatorGui.java ×
1 package swing;
2
3 import java.awt.*;
4 import javax.swing.JFrame;
5
6 public class CalculatorGui extends JFrame{
7
8     public CalculatorGui(String title) {
9         super(title);
10        setVisible(true);
11        setLayout(null);
12        Font font = new Font("TimesRoman", Font.PLAIN, 14);
13        setFont(font);
14        getContentPane().setBackground(Color.blue);
15        setSize(300,350);
16        setLocation(40,80);
17        setResizable(false);
18
19        setVisible(true);
20        setDefaultCloseOperation(EXIT_ON_CLOSE);
21    }
22 }

```

Σχήμα 6.4 Ορισμός της κλάσης CalculatorGui για το iteration1 με χρήση της swing.

## 6.4.2 Iteration No 2 – Προσθήκη Button και ορισμός συμπεριφοράς

**E3.3.** Μετά την γραμμή κώδικα 11 της κλάσης CalculatorGui προσθέστε τον κώδικα που δημιουργεί ένα στιγμιότυπο της κλάσης Button και ορίζει τα χαρακτηριστικά του. Αν χρησιμοποιήσατε swing θα χρησιμοποιήσετε την JButton.

```

14 button0 = new Button("0");
15 button0.setBounds(64, 265, 35, 28);
16 button0.setFont(new Font("TimesRoman", Font.PLAIN, 14));
17 button0.setForeground(Color.blue);

```

Τρέξτε το πρόγραμμά σας. Δεν θα δείτε το button στη γραφική διεπαφή και αυτό γιατί δεν ενημερώσατε για αυτό το στιγμιότυπο της κλάσης CalculatorGui.

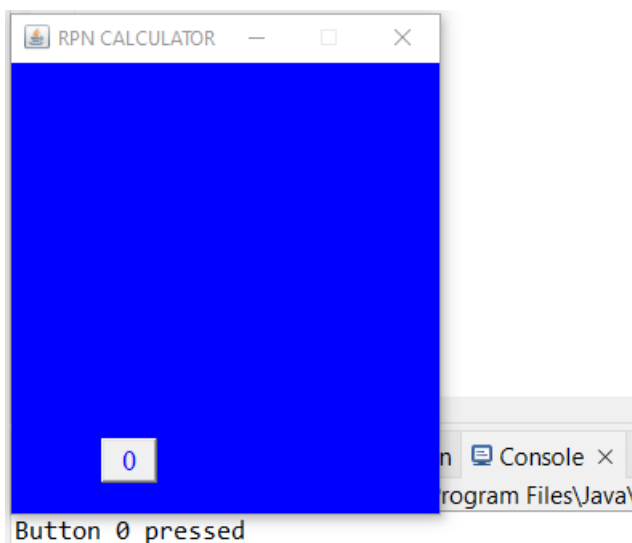
**E3.4.** Αυτό θα το κάνετε προσθέτοντάς την παρακάτω πρόταση.

```
this.add(button0);
```

Τρέξτε το πρόγραμμα σας. Τώρα θα δείτε το button. Πιέστε το. Δεν υπάρχει καμιά απόκριση από το πρόγραμμα σας. Απλά γιατί δεν έχετε ορίσει απόκριση για το συμβάν πίεσης του πλήκτρου.

**E3.5.** Για να ορίσετε απόκριση θα αξιοποιήσετε τον μηχανισμό του event handling. Θα χρειασθεί να ανατρέξετε στις διαφάνειες της αντίστοιχης διάλεξης για να δείτε πως θα ορίσετε τον event handler και πως θα κάνετε register ένα στιγμιότυπο του στο button0 με την μέθοδο addActionListener(). Ορίστε στον event handler ως ενέργεια να τυπώνει στην οθόνη το μήνυμα "Button 0 pressed".

Τρέξτε το πρόγραμμα σας. Θα δείτε την γραφική διεπαφή που δείχνει το Σχήμα 6.5. Στην console έχει τυπωθεί το μήνυμα σας (κάτω από την γραφική διεπαφή).



*Σχήμα 6.5 Η γραφική διεπαφή με το Button και ορισμένη συμπεριφορά για αυτό.*

Στο επόμενο iteration θα προσθέσουμε στη γραφική διεπαφή το display.

### 6.4.3 Iteration No 3 – Προσθήκη display

**E3.6.** Δημιουργήστε το display ως στιγμιότυπο της TextField και ορίστε τα χαρακτηριστικά του. Το δίνει τον κώδικα που κάνει το έργο αυτό. Αν χρησιμοποιήσατε swing θα χρησιμοποιήσετε την JTextField.



```
22 display = new TextField("0");
23 display.setEditable(false);
24 display.setBounds(13, 55, 257, 30);
```

Σχήμα 6.6 Δημιουργία του `display` της γραφικής διεπαφής και ορισμός χαρακτηριστικών του.

Προσέξτε, για να το δείτε χρειάζεται να κάνετε ακόμη μια ενέργεια. Την γνωρίζετε. Θα πρέπει να την γνωρίζετε!

**E3.7.** Αφού τροποποιήσετε τον κώδικα σας ώστε να εμφανίζεται το `display` τροποποιήστε τον κώδικα σας ώστε όταν πατάτε το `button0` να εμφανίζεται το μήνυμα `"Button 0 pressed"` στο `display` της γραφικής διεπαφής.

Όταν το πετύχετε αυτό, θα έχετε ολοκληρώσει μια έκδοση της γραφικής διεπαφής που έχει τα βασικά που θα σας επιτρέψουν σε ένα επόμενο βήμα να την εξελίξετε ώστε να σας επιτρέψει τον υπολογισμό της τιμής μιας απλής έκφρασης.

#### 6.4.4 Iteration No 4 – Προσθήκη της Operand

**E3.8.** Προσθέστε την κλάση `Operand` την οποία έχετε ήδη έτοιμη από προηγούμενη δραστηριότητα. Τροποποιήστε κατάλληλα τον κώδικα σας ώστε με το πάτημα του πλήκτρου `button0` να αποστέλλεται μήνυμα `addDigit()` στο στιγμιότυπο της `Operand`. Προσέξτε πως για να γίνει αυτό ο `event handler` θα πρέπει να μπορεί να αναφερθεί στο στιγμιότυπο της `Operand`.

**E3.9.** Τροποποιήστε την `Operand` ώστε να σας εμφανίζει την τρέχουσα τιμή της στο `display` της γραφικής διεπαφής.

**E3.10** Προσθέστε στη γραφική σας διεπαφή ένα ακόμη πλήκτρο, το `button1`. Τρέξτε το πρόγραμμά σας και ελέγξτε την σωστή λειτουργία του.

#### 6.4.5 Iteration No 5 – Προσθήκη των Adder και ResultPresenter

**E3.11.** Διαμορφώστε τη γραφική διεπαφή ώστε να έχει τα πλήκτρα 1, ENTER, + και =.

**E3.12.** Ορίστε κατάλληλα τη συμπεριφορά των πλήκτρων ENTER, + και =. Θα έχετε μια αριθμομηχανή με την οποία θα μπορείτε να υπολογίσετε την τιμή της έκφρασης `100 10 + =`.

## 7. Χειρισμός εξαιρέσεων

Στην τελευταία έκδοση της RPNCalculator που αναπτύξατε πιέστε το πλήκτρο + χωρίς να έχετε εισάγει τελεστέο. Θα δείτε πως το σύστημα τυπώνει μήνυμα όπως το παρακάτω

```
Exception in thread "AWT-EventQueue-0" java.util.EmptyStackException
```

Αυτό συμβαίνει γιατί δεν συλλάβατε εσείς την εξαίρεση EmptyStackException ώστε να την διαχειριστείτε με τον τρόπο που εσείς θέλετε.

### Δράση 7.1 – Προσθέστε χειρισμό εξαιρέσεων

1. Τροποποιήστε τον κώδικα σας ώστε σε περίπτωση έγερσης της εξαίρεσης EmptyStackException να τυπώνεται κατάλληλο μήνυμα, π.χ. Stack Empty, στο display της γραφικής διεπαφής της αριθμομηχανής σας.
2. Ελέγξτε τη συμπεριφορά της αριθμομηχανής σας για διαίρεση με το 0. Τροποποιήστε τον κώδικα ώστε όταν ο χρήστης κάνει προσπάθεια για διαίρεση με 0 να εμφανίζεται στην οθόνη κατάλληλο μήνυμα (π.χ. Attempt to divide by zero).

## 8. Βελτιώσεις ποιότητας κώδικα

### Δράση 8.1 – Βελτίωση Ποιότητας κώδικα

Αξιοποιήστε τις γνώσεις σας για να βελτιώσετε την ποιότητα του κώδικα της εφαρμογής σας. Για παράδειγμα:

- Αν χρησιμοποιείτε στην εφαρμογή σας γενικές μεταβλητές, κάντε προσπάθεια να τις αποφύγετε. Υπάρχει τρόπος.
- Αν στην εφαρμογή σας χρησιμοποιείτε την πρόταση switch προσπαθήστε να δώσετε κώδικα χωρίς την χρήση της. Τις περισσότερες φορές υπάρχει τρόπος.