Εργαστήριο 2 - Άσκηση - Ανάλυση

Εκφώνηση: Δημιουργείστε την εφαρμογή Αριθμήτηριο σε Java Swing με χρήση NetBeans ακολουθώντας τις παρακάτω οδηγίες.

- Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame, ο οποίος θα έχει τίτλο «Calculator», διάταξη GridBagLayout και θα οριστεί με την έξοδο του να κλείνει και η εφαρμογή.
- 2. Τοποθετήστε στο JFrame ένα TextField και 18 Buttons.
- 3. Τοποθετήστε τα συστατικά όπως φαίνονται στην παρακάτω εικόνα με χρήση του εργαλείου 'Customize Layout' (δεξί click πάνω στο φόντο).
- 4. Αλλάξτε τις ιδιότητες του TextField ως εξής:
 - Variable name: calcbox
 - Font: bold 24 μέγεθος.
 - Horizontal Alignment: Right
- 5. Αλλάξτε τις ιδιότητες των 18 Buttons ως εξής:
 - Font: bold 18 μέγεθος
 - το χρώμα το κουμπιών να ταιριάζει με την παρακάτω εικόνα:

4	Cal	culator	-	×
	POV	VER		С
	7	8	9	+
	4	5	6	-
	1	2	3	*
	0		=	1

6. Να προγραμματιστεί το κουμπί POWER ώστε να ενεργοποιεί ή απενεργοποιεί όλα τα άλλα συστατικά με χρήση της μεθόδου setEnabled(Boolean). Αρχικά όλα τα συστατικά να είναι απενεργοποιημένα όπως φαίνονται στην παρακάτω εικόνα:

≝	Cal	culator	-	×	
ſ					
	POV	VER		С	
	7	8	9	+	
	4	5	6	-	
	1	2	3	*	
	0		=	/	

- Να προγραμματιστούν τα δέκα κουμπιά αριθμών ώστε να προσθέτουν τον αντίστοιχο αριθμό κάθε κουμπιού δεξιά του περιεχομένου του calcbox.
- 8. Να προγραμματιστεί το κουμπί «.» ώστε να μπαίνει η υποδιαστολή στο **calcbox.** Προσοχή να μην μπαίνει δεύτερη υποδιαστολή στον ίδιο αριθμό.
- 9. Να προγραμματιστούν τα τέσσερα κουμπιά των πράξεων για να υλοποιούν τις αντίστοιχες πράξεις.
- 10. Να προγραμματιστεί το κουμπί ίσον ώστε να δίνει αποτέλεσμα:
 - 0, αν πατηθεί το ίσον πριν την εισαγωγή κάποιου αριθμού,
 - ο πρώτος αριθμός που εισήχθη, εάν μετά το πάτημα ενός συμβόλου πράξης πατηθεί το ίσον χωρίς εισαγωγή άλλου αριθμού,
 - μήνυμα λάθους, αν σε διαίρεση ο δεύτερος αριθμός είναι το 0,
 - το αποτέλεσμα της πράξης σε κάθε άλλη περίπτωση.
- 11. Να προγραμματιστεί το κουμπί C ώστε να αδειάζει το περιεχόμενο του **calcbox** και να μηδενίζει τις μεταβλητές που θα χρησιμοποιήσετε για τις πράξεις.
- 12. Να προγραμματιστεί το πλήκτρο <ENTER> στο συστατικό κειμένου ώστε να έχει την ίδια λειτουργία με το ίσον.

Σημείωση: Μέθοδοι που πιθανώς θα σας φανούν χρήσιμες:

- JTextField.setText(string);
- JTextField.getText();
- JComponent.setEnabled(Boolean);
- JComponent.isEnabled();
- Πεδίο_String.isEmpty();
- String.valueOf(double);
- Double.valueOf(string);
- Πεδίο_String.contentEquals(string)
- getRootPane().setDefaultButton(action);

Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi4.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio2.

Βήμα 3: Ορισμός Ιδιοτήτων στο JFrame. Ορίζουμε ως τίτλο του JFrame το «Calculator».

Βήμα 4: Ορίζουμε η διάταξη του JFrame να είναι GridBagLayout. Για να ορίσουμε τη διάταξη ανοίγουμε το μενού του JFrame επιλέγοντας το δεξί πλήκτρο του ποντικιού πάνω στο JFrame. Στη συνέχεια επιλέγουμε το υπομενού «Set Layout» και τη διάταξη «GridBagLayout» - Εικόνα 1. Όταν ορίσουμε τη διάταξη, αυτή θα εμφανιστεί στο *Navigator* κάτω ακριβώς από το JFrame – Εικόνα 2.



Εικόνα 1: Ορισμός διάταξης GridBagLayout



Εικόνα 2: Στο Navigator παρουσιάζεται η διάταξη που ορίσαμε

Βήμα 5: Τοποθετούμε ένα TextField και 18 Buttons με drag & drop - Εικόνα 3.

Projects × Files Services -	Ergastirio2.java ×	□ Palette ×	-
🗈 🆢 Askisi1 🔨 🔨	Source Design History	Swing Containers	^
🗈 🎒 Askisi2		Panel	Tabbed Pane
🕀 🍥 Askisi3		Split Pane	Scroll Pane
🗄 🍅 Askisi4		Tool Bar	
🖶 🏠 Source Packages			
efault package>		Internal Frame	Layered Pane
Ergastirio2.java		Swing Controls	
Test Packages		label Label	OK Button
Libranes		I Toggle Button	E Check Box
jButton18 [JButton] - Navigator × -	Butten 1 Butten 2 Butten 2 Butten 1 But		8 Button Group
	parcenz joarceus joarceus joarceus joarceus joarceus joarceus joarceus joarceus	Combo Box	E List
Ender Other Components		Tayt Field	tx Text Area
CridBad avout		Text field	
TextField1 [TextField]		Mill Scroll Bar	uju Sider
Button 1 (Button)		Progress Bar	Formatted Field
IButton2 [JButton]		iButton18 [JButton] - Pro	perties X -
IButton3 [JButton]		Properties Binding	Evente Code
jButton4 [JButton]		Properces binding	Evena cooc
jButton5 [JButton]		Properties	-
jButton6 [JButton]		action	Churez
jButton7 [JButton]		foot	Tahoma 11 Plain
jButton8 [JButton]		foreground	
··· 💷 jButton9 [JButton]		icon	<none></none>
··· 📧 jButton 10 [JButton]		mnemonic	
jButton 11 [JButton]		text	jButton 18
jButton12 [JButton]		toolTipText	
jButton13 [JButton]		Other Properties	
DE jButton 14 [JButton]		UIClassID	ButtonUI ¥
Button 15 [JButton]		iButton18 [IButton]	0
(IV) (Button 17 [Button]		Justice (Sourcon)	•
(iii) joction 2 (Joutton)			
in portonio poution			I This

Εικόνα 3: Τοποθετήθηκε ένα Textfield και δεκαοχτώ Button

Βήμα 6: Αλλάζουμε το κείμενο που εμφανίζουν τα κουμπιά σύμφωνα με την εκφώνηση, δηλαδή 'POWER', 'C', 'O', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.', '+', '-', '*', '/' και '=' και τα ονόματα των μεταβλητών τους αντίστοιχα σε 'Bpower', 'BC', 'BO', 'B1', 'B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B9', 'Bdot', 'Bplus', 'Bminus', 'Btimes', 'Bdivide' και 'Bequal'.

Projects × Files Services -	📑 Ergastirio2.java 🗙 🔶 🕚	Palette ×		-
🕒 🎂 Askisi1 🔷 🔨	Source Design History	😨 😑 Swing Containe	rs	^
🗄 🍥 🍅 Askisi2		Panel	Tabbed Pane	
🕀 🍅 Askisi3		Split Pane	Scroll Pane	
B-S Askisi4		I Tool Bar	📇 Desktop Pane	
General source Packages		Internal Frame	Layered Pane	
Ergastirio2.java		Swing Controls	<u> </u>	
- Test Packages		label Label	GK Button	
🕀 🔓 Libraries		IN Tooole Button	E- Check Box	
		R- Radio Rutton	8- Rutton Crown	
jButton18 [JButton] - Navigator × -	TextField1 POWER C 0 1 2 3 4 5 6 7 8 9 . + - * / =	- Kaulo battori		
Other Components		Compo Box	En List	
e- [JFrame]		Text Field	Text Area	
GradbagLayout		Scroll Bar	💭 Sider	
Bower [Button]		Progress Bar	Formatted Field	~
BC [Button]		Requel [Putten]	Properties Y	
B0 [JButton]		Dequal [Jourton] -	Froperties A	
B1 [JButton]		Properdes Bridin	ig Events Code	
B2 [JButton]		Properties		^
DK B3 [JButton]		action	<none></none>	
DK B4 [JButton]		background	[240,240,240]	
B5 [JButton]		tont	Tanoma 11 Plain	
B6 [JButton]		foreground	[0,0,0]	
DK B7 [JButton]		icon	<none> v</none>	
DK B8 [JButton]		mnemonic		
DK B9 [JButton]		text		
Bdot [JButton]		toolliplext		
DE Bplus [JButton]		Coner Properties	Datast	
DK Bminus [JButton]		UICIASSID	Buttonul	-
DK Btimes [JButton]		Bequal [JButton]	(3
Bdivide [JButton]				
Bequal [JButton]				
			1	TAIC

Εικόνα 4: Ορίστηκαν οι ιδιότητες text και variable name για τα δεκαοχτώ κουμπιά

Βήμα 7: Για να τοποθετήσουμε τα συστατικά στη μορφή που ζητάει η εκφώνηση θα χρησιμοποιήσουμε το εργαλείο "Customize Layout". Για να χρησιμοποιήσουμε το εργαλείο "Customize Layout" θα πρέπει να το «ανοίξουμε» με δεξί click πάνω στο JFrame, δηλαδή στον καμβά (Εικόνα 5). Στην Εικόνα 6 παρουσιάζεται το εργαλείο με την αρχική τοποθέτηση των συστατικών. Στη συνέχεια με drag & drop μετακινούμε τα συστατικά στη θέση τους, όπως φαίνεται στην Εικόνα 7. Για να *καταλάβει* το συστατικό POWER δύο κελιά επιλέγουμε το συστατικό και στη συνέχεια πατάμε το κουμπί Horizontal Size in Grid (Εικόνα 8). Για να απλωθεί το συστατικό σε όλο το πλάτος των κελιών επιλέγουμε το συστατικό και στη συνέχεια πατάμε το κουμπί Horizontal Fill (Εικόνα 9). Χρησιμοποιούμε το Horizontal Size in Grid στο συστατικό jTextField1, ώστε να καταλάβει 4 κελιά. Εφαρμόζουμε το Horizontal Fill σε όλα τα συστατικά. Στην Εικόνα 10, φαίνεται η τελική τοποθέτηση των συστατικών μετά την εφαρμογή όλων των παραπάνω. Κλείνουμε το εργαλείο για να προχωρήσουμε στο επόμενο βήμα.

Start Page × Ergastirio2.java ×	· · · · · · · · · · · · · · · · · · ·	•
Source Design History	⇔ \$	8
TextField1 POWER C 0 1 2 3 4	Preview Design > Bind > Events > Set Layout > Customize Layout + - / Add From Palette > Design This Container > Set to Default Size Edit Layout Space Change Order C	-
	Paste Ctrl+V Customize Code	
	Properties	





Εικόνα 6: Το εργαλείο Customize Layout – Τα συστατικά στην αρχική τους θέση



Εικόνα 7: Τα συστατικά στην τελική τους θέση



Εικόνα 8: Χρήση του Horizontal Size in Grid στο συστατικό POWER

Anchor	Insets	50	# *	5 🛟	*	5 🗘	6	
र के म			0	1	2	3		
4 . *		and a						
M 47 M		0	Text-leid1		0			
	4	1 9	POWE	R	•	С		
Internal Dauldan		2	7	8	9	+		
Internal Pabling		3	4	5	6	•		
		4	1	2	3	8		
		5	0		-	1		
	5	10000		1	0.23			
Size in Grid								
	Fil (1987)							
* + + + +	P 23							
Position in Grid	Weights Horizontal Fill							
	41 CP 00 00 000							
Q Q								
Layout Constraints								
irid X	0	~						
irid Y	1	~						
irid Width	2	~						
Brid Height	1	~						
ill	Honzontal	~						
iternal Padding X	0							
Iternal Padding Y	0							
								>

Εικόνα 9: Χρήση του Horizontal Fill στο συστατικό POWER



Εικόνα 10: Η τελική τοποθέτηση των συστατικών

Βήμα 8: Αλλάζουμε το όνομα μεταβλητής του *jTextField1* σε *calcbox*. Στη συνέχεια αλλάζουμε τρεις ιδιότητες του *calcbox* (Εικόνα 11). Την ιδιότητα *font* για να ορίσουμε μέγεθος γραμματοσειράς 24 και στυλ *bold*, την ιδιότητα *horizontalAlignment* σε *RIGHT* για δεξιά στοίχιση του περιεχομένου του TextField και την ιδιότητα *text* σε κενό – σβήνουμε το περιεχόμενο. Σε περίπτωση που δεν θέλουμε να μπορεί ο χρήστης να γράψει αριθμούς στο TextField με το πληκτρολόγιο, ορίζουμε ανενεργή την ιδιότητα *editable*.

				Properties Binding Events	Code		
				Properties			
				editable			
				background	[255,255,255]		
			_	columns	0		
				document	<default> ~</default>		
-			- (font	Tahoma 24 Bold		
POWER C				foreground	[0,0,0]		
7	8	9	+	horizontalAlignment	RIGHT V		
4	5	6		text			
			toolTipText				
1	2	3		Other Properties			
0	•	-	1	UI	<default></default>		
				UIClassID	TextFieldUI		
				action	<none> ~</none>		
				alignmentX	0.5		
				alignmentY	0.5		
				autoscrolls			
				baselineResizeBehavior	CENTER_OFFSET		
				border	[XPFillBorder]		
				caret	<default></default>		
				caretColor	[0,0,0]		
				caretPosition	0		
					· · ·		

Εικόνα 11: Ορισμός ιδιοτήτων για το TextField

Βήμα 9: Επιλέγουμε τα 18 JButton και αλλάζουμε την ιδιότητα *font* όπως αναφέρεται στην εκφώνηση – μέγεθος 18, στυλ bold. Επιλέγουμε τα JButton που σύμφωνα με την εκφώνηση πρέπει να γίνουν κόκκινα, και αλλάζουμε την ιδιότητα *foreground* σε κόκκινο. Στην Εικόνα 12, παρουσιάζεται πως διαμορφώνεται η εικόνα της εφαρμογής μετά από τις παραπάνω αλλαγές.

С		WER	PO	
+	9	7 8		
-	6	4 5		
*	3	2	1	
1	=		0	
C + - *	9 6 3	8 5 2	PO 7 4 1	

Εικόνα 12: Η εικόνα της εφαρμογής μετά την εφαρμογή του βήματος 9.

Βήμα 10: Σε αυτό το βήμα θα φροντίσουμε όλα τα συστατικά του καμβά εκτός του κουμπιού Power να «ξεκινούν» απενεργοποιημένα. Αυτό γίνεται ορίζοντας την τιμή false στην ιδιότητα *enabled* όλων των συστατικών - Εικόνα 13.

Source Design	History	IC 🔛	1			6-4-	
					Properties Binding Events	Code	
					Properties		
					background	[240,240,240]	
					font	<different values=""></different>	
					foreground	<different values=""></different>	
					text	<different values=""></different>	
				-	tooITipText	<different values=""></different>	
	PO	WER		C	Other Properties		
	7	8	0	- A -	UIClassID	<different ∀alues=""></different>	
		0	9	Ŧ	alignmentX	<different values=""></different>	
	4	5	6	- 1	alignmentY	0.5	
	-				autoscrolls		
	1	2	3	*	baselineResizeBehavior	CENTER_OFFSET	
			border	<different values=""></different>			
	0	•	=	1	componentPopupMenu	<none></none>	~
					cursor	Default Cursor	~
					debugGraphicsOptions	NO_CHANGES	~
					doubleBuffered		
					enabled		
					focusCycleRoot		
					focusTraversalPolicy	<none></none>	~
					focusTraversalPolicyProvider		[
					focusable		[.
					enabled		
utput ×					(boolean) The enabled state of the co	omponent.	

Εικόνα 13: Απενεργοποίηση των συστατικών

Εναλλακτικά, ένας δεύτερος τρόπος που μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με κώδικα, φαίνεται στην Εικόνα 14. Αρχικά, καλούμε στον δομητή της τάξης, υποχρεωτικά μετά τη μέθοδο *initComponents()*, τη δική μας μέθοδο *disable_components()*. Η δήλωση της μεθόδου δίνεται στις γραμμές 12-17. Η μέθοδος δηλώνεται ως void. Στη γραμμή 13-14 ορίζεται ένας πίνακας από *Component* – είναι απαραίτητη η εισαγωγή της βιβλιοθήκης *java.awt.Component* (γραμμή 2). Προφανώς, στον πίνακα θα δηλώσουμε όλα τα συστατικά του καμβά εκτός του *Bpower*, το οποίο είναι πάντα ενεργό. Στη συνέχεια στις γραμμές 15-16 ορίζεται η ιδιότητα enabled στη τιμή false με τη μέθοδο *setEnabled(Boolean)* για όλα τα συστατικά του πίνακα *CalcComponents*.

🖒 Erg	jastiri	o2.java ×
Sourc	e (Design History 📴 🖓 - 🐺 - 💐 🖓 🖓 🚭 🎧 🌮 😓 🖓 🗐 🗐 🗐 😑 🔲 🖉 🚅
1		
2	-	<pre>import java.awt.Component;</pre>
3		
4		public class Ergastirio2 extends javax.swing.Jrrame {
6		<pre>@SuppressWarnings("OverridableMethodCallInConstructor")</pre>
7	-	<pre>public Ergastirio2() {</pre>
8		<pre>initComponents();</pre>
9		disable_components();
10	L	}
11		
12	무	void disable components() {
13		Component[] Calcomponents = (BU, B1, B2, B3, B4, B5, B6, B7, B8, B9, BC,
14		Balvide, Baot, Bequal, Bminus, Bplus, Btimes, Calcbox};
15		for (Component comp : Calcomponents)
16		<pre>comp.setEnabled(false);</pre>
17		2
18		
19		
20	_	(SuppressWarnings ("unchecked")
21	+	Generated Code

Εικόνα 14: Απενεργοποίηση των συστατικών με κώδικα

Ένας τρίτος τρόπος, πάλι με κώδικα φαίνεται στην Εικόνα 15. Η διαφορά από τον προηγούμενο τρόπο είναι ότι ο πίνακας των συστατικών δημιουργείται δυναμικά. Έτσι, στη γραμμή 13, αποθηκεύεται στη μεταβλητή compcount ο αριθμός των συστατικών που υπάρχουν στον καμβά μας με χρήση της μεθόδου getComponentCount(). Η getContentPane() μέθοδος επιστρέφει τον τρέχον υποδοχέα. Για το παράδειγμά μας είναι ο καμβάς, το JFrame. Στη συνέχεια, γραμμές 14-18, σε ένα κόμβο for, με την getContentPane().getComponent(i) διαβάζει τη μεταβλητή κάθε συστατικού του καμβά, την αποθηκεύει στην μεταβλητή comp, ελέγχει αν δεν είναι το Bpower, και απενεργοποιεί το αντίστοιχο συστατικό.

```
🗟 Ergastirio2.java 🛛 🗙
Source Design History 🛛 🚱 🗸 🚚 🗸 🔁 🖓 😓 🎝 🖓 😓 🖓 😓 🖓 😓 🖓
  1
  2
    import java.awt.Component;
  3
       public class Ergastirio2 extends javax.swing.JFrame {
  4
  5
           @SuppressWarnings("OverridableMethodCallInConstructor")
  6
  7
    _
           public Ergastirio2() {
  8
               initComponents();
  9
               disable_components();
 10
           3
 11
 12 -
           void disable_components() {
 13
               int compcount = getContentPane().getComponentCount();
 14
               for (int i = 0; i < compcount; i++) {</pre>
 15
                   Component comp = getContentPane().getComponent(i);
                   if (comp != Bpower)
 16
 17
                       comp.setEnabled(false);
 18
                }
 19
           3
 20
 21
           @SuppressWarnings("unchecked")
 22 +
           Generated Code
```

Εικόνα 15: Δημιουργία του πίνακα συστατικών δυναμικά

Βήμα 11: Το κουμπί *Bpower* σύμφωνα με την εκφώνηση θα πρέπει να αλλάζει κατάσταση στα υπόλοιπα συστατικά του καμβά. Αυτό σημαίνει ότι αν τα συστατικά είναι απενεργοποιημένα θα τα ενεργοποιεί και το αντίστροφο. Εννοείται πως το κουμπί *Bpower* θα είναι πάντα ενεργό. Έτσι, δημιουργούμε ένα *Action* γεγονός για το κουμπί *Bpower*. Με βάση αυτά που μελετήσαμε στο προηγούμενο βήμα θα δούμε τέσσερις τρόπους υλοποίησης της ενέργειας *ActionPerformed* του κουμπιού *Bpower*.

Ο πρώτος τρόπος φαίνεται στην Εικόνα 16. Στη γραμμή 221 με χρήση της μεθόδου isEnabled() ελέγχουμε αν ένα συστατικό είναι ενεργό. Μπορούμε να κάνουμε τον έλεγχο με βάση οποιοδήποτε συστατικό εκτός του *Bpower* που είναι πάντα ενεργό. Αν είναι ενεργό το συστατικό (αν ο έλεγχος *BC.isEnabled()* επιστρέψει *true*) τότε απενεργοποιούμε όλα τα συστατικά – γραμμές 222-239, αλλιώς τα ενεργοποιούμε – γραμμές 241-258.

🚮 Ergastirio2.ja	va ×
Source Design	History 🛛 🔞 🗸 🕄 😓 😓 🔂 🖓 😓 😓 🖄 의 의 🕘 🔐 🚅
22 L	Senerateu coue
219	
220 -	<pre>private void BpowerActionPerformed(]ava.awt.event.ActionEvent evt) { if (PC i=Earblad())(</pre>
221	II (BC.1SENADICO()) {
222	BC.SetEnabled(false);
223	D0.SetEnabled(false);
224	B1.SetEnabled(false);
225	D2.setEnabled(false);
220	B3.SetEnabled(false);
227	D4.SetEnaDied(Talse);
228	DO.SetEnaDied(Talse);
229	B0.SetEnabled(false);
230	B/.SetEnabled(false);
231	P0 actProbled(false);
232	Pdivide setProbled(false);
233	Edit atPabled(false);
231	Borryal actEmphied (false);
235	Eminus setEnabled (false);
230	Btimes setEnabled(false);
238	Enlug setEnabled(falge):
239	calchox setEnabled(false):
240	} also {
241	BC setEnabled(true).
242	B0.setEnabled(true):
243	B1.setEnabled(true):
244	B2.setEnabled(true):
245	B3.setEnabled(true);
246	P4 actEnabled (true)
247	P5. gotEnabled(true);
248	B6 setEnabled(true);
249	B7 setEnabled(true).
250	B8 setEnabled(true).
251	B9 setEnabled(true).
252	Bdivide setEnabled(true).
253	<pre>Bdot.setEnabled(true):</pre>
254	Begual setEnabled(true).
255	Bminus.setEnabled(true):
256	Btimes setEnabled(true).
257	Bolus setEnabled(true).
258	calchow setEnabled(true).
250	Carobox.setEnableu(true);
260	
260	1
261	/** 2 lines */
202 1	mublic static moid main(Stwing area[]) [21 lines]
203 🖽	public static void main(string args[]) {si lines }

Εικόνα 16: Υλοποίηση της ενέργειας ActionPerformed του κουμπιού Bpower – 1^{ος} τρόπος

Ο δεύτερος τρόπος φαίνεται στην Εικόνα 17. Το κουμπί Bpower υλοποιείται με τη λογική να αντιστρέφει την κατάσταση των κουμπιών. Έτσι, με την εντολή *BC.setEnabled(!BC.isEnabled())*, η παράμετρος γίνεται false όταν το *BC.isEnabled()* επιστρέφει true (με το !) και το αντίστροφο.

📑 Erga	stirio	2.java ×	
Source	D	esign Hist	ory 📴 😼 - 💭 - 🔍 👎 🖶 斗 🍄 😓 🧏 🥶 🖄 👄 💷 🕮 🚅
22	+	Gen	erated Code
219			
220	Ę	priv	<pre>vate void BpowerActionPerformed(java.awt.event.ActionEvent evt) {</pre>
221			<pre>BC.setEnabled(!BC.isEnabled());</pre>
222			<pre>B0.setEnabled(!B0.isEnabled());</pre>
223			<pre>B1.setEnabled(!B1.isEnabled());</pre>
224			<pre>B2.setEnabled(!B2.isEnabled());</pre>
225			<pre>B3.setEnabled(!B3.isEnabled());</pre>
226			<pre>B4.setEnabled(!B4.isEnabled());</pre>
227			<pre>B5.setEnabled(!B5.isEnabled());</pre>
228			<pre>B6.setEnabled(!B6.isEnabled());</pre>
229			<pre>B7.setEnabled(!B7.isEnabled());</pre>
230			<pre>B8.setEnabled(!B8.isEnabled());</pre>
231			<pre>B9.setEnabled(!B9.isEnabled());</pre>
232			<pre>Bdivide.setEnabled(!Bdivide.isEnabled());</pre>
233			<pre>Bdot.setEnabled(!Bdot.isEnabled());</pre>
234			<pre>Bequal.setEnabled(!Bequal.isEnabled());</pre>
235			<pre>Bminus.setEnabled(!Bminus.isEnabled());</pre>
236			<pre>Btimes.setEnabled(!Btimes.isEnabled());</pre>
237			<pre>Bplus.setEnabled(!Bplus.isEnabled());</pre>
238			<pre>calcbox.setEnabled(!calcbox.isEnabled());</pre>
239	L	}	
240			

Εικόνα 17: Υλοποίηση της ενέργειας ActionPerformed του κουμπιού Bpower – 2°ς τρόπος

Ο τρίτος τρόπος φαίνεται στην Εικόνα 18. Η προσέγγισή του βασίζεται στη δήλωση του πίνακα συστατικών *CalcComponents*, ο οποίος περιέχει όλα τα συστατικά που αλλάζουν κατάσταση.

📄 Erg	astirio	2.java	X
Source	e D	esign	History 📴 🗟 + 👼 + 💐 🞝 🞝 🞝 🚭 斗 🤗 😓 🖓 🖄 😓 🗐 🖄 🥚 🗐 🏥 🚅
22	+		Generated Code
219			
220	Ę		private void BpowerActionPerformed(java.awt.event.ActionEvent evt) {
221			Component[] CalcComponents = {B0, B1, B2, B3, B4, B5, B6, B7, B8, B9,
222			<pre>BC, Bdivide, Bdot, Bequal, Bminus, Btimes, Bplus, calcbox};</pre>
223			<pre>for (Component comp : CalcComponents)</pre>
224			<pre>comp.setEnabled(!comp.isEnabled());</pre>
225	L		}
226			
227	+		/**3 lines */
230	+		<pre>public static void main(String args[]) {31 lines }</pre>
261			



Ο τέταρτος τρόπος φαίνεται στην Εικόνα 19. Η λογική της υλοποίησής του βασίζεται στη δημιουργία του πίνακα συστατικών με δυναμικό τρόπο.

🔂 Ergastirio2. java 🛛 🗙		
Source	Desig	gn History 📴 😼 = 💭 = 🛛 🕄 🖓 😓 🖓 😓 🖓 🗐 🗐 🗐 🥌 🗐 🏥 🚅
22	+	Generated Code
219		
220	Ę	<pre>private void BpowerActionPerformed(java.awt.event.ActionEvent evt) {</pre>
221		<pre>int compcount = getContentPane().getComponentCount();</pre>
222		<pre>for (int i = 0; i < compcount; i++) {</pre>
223		Component comp = getContentPane().getComponent(i);
224		<pre>if (comp != Bpower)</pre>
225		<pre>comp.setEnabled(!comp.isEnabled());</pre>
226		}
227	L	}
228		
229	+	/**3 lines */
232	÷	<pre>public static void main(String args[]) {31 lines }</pre>
263		

Εικόνα 19: Υλοποίηση της ενέργειας ActionPerformed του κουμπιού Bpower – 4°ς τρόπος

Βήμα 12: Στη συνέχεια θα προγραμματίσουμε το *Action* γεγονός των δέκα αριθμών κουμπιών ώστε όταν τα επιλέγει ο χρήστης να γράφεται ένας αριθμός δεξιά των αριθμών που έχουν γραφτεί στο *calcbox*. Έτσι, αφού δημιουργήσουμε το *Action* γεγονός ενός κουμπιού υλοποιούμε την ενέργεια *ActionPerformed*, όπως φαίνεται στην Εικόνα 20. Για να τοποθετηθεί ο αριθμός που επιλέγουμε δεξιά αυτών που υπάρχουν ήδη στο calcbox θα πρέπει πρώτα να διαβάσουμε το περιεχόμενο του calcbox με τη μέθοδο getText() και στη συνέχεια να προσθέσουμε (με string concatenation) τον αριθμό.

👔 Ergastirio2.java 🛛 🗙			
Source	e D	esign History 🞯 🔯 = 🚚 = 🔍 🖏 🖓 🖶 🏹 🍄 😓 🏷 🗐 🗐 😑 🔲 🗳 🚅	
279	Ę	<pre>private void BOActionPerformed(java.awt.event.ActionEvent evt) {</pre>	
280		<pre>calcbox.setText(calcbox.getText()+"0");</pre>	
281	L	}	
282			
283	F	<pre>private void BlActionPerformed(java.awt.event.ActionEvent evt) {</pre>	
284		<pre>calcbox.setText(calcbox.getText()+"1");</pre>	
285	L	}	
286	_		
287	P	<pre>private void B2ActionPerformed(java.awt.event.ActionEvent evt) {</pre>	
288		<pre>calcbox.setText(calcbox.getText()+"2");</pre>	
289			
290			
291	뷔	private void B3ActionPerformed(java.awt.event.ActionEvent evt) {	
292		calcbox.setText(calcbox.getText()+"3");	
293		}	
294		private word BADationBenformed/java sut event ActionEvent evt)	
295	Т	calchox setText (calchox setText ()+"4");	
297	L	}	
298		1	
299		private void B5ActionPerformed(java.awt.event.ActionEvent evt) {	
300	Τ	calcbox.setText(calcbox.getText()+"5");	
301	L	}	
302			
303	Ę	<pre>private void B6ActionPerformed(java.awt.event.ActionEvent evt) {</pre>	
304		<pre>calcbox.setText(calcbox.getText()+"6");</pre>	
305	L	}	
306	_		
307	F	private void B7ActionPerformed(java.awt.event.ActionEvent evt) {	
308		<pre>calcbox.setText(calcbox.getText()+"/");</pre>	
309	_	}	
310		private word B93 ation Depformed (java sut event lation Front out) (
312	Ť	calchox setText (calchox getText()+"8"):	
313	L	}	
314		,	
315	Ξ	private void B9ActionPerformed(java.awt.event.ActionEvent evt) {	
316		<pre>calcbox.setText(calcbox.getText()+"9");</pre>	
317	L	}	
318			

Εικόνα 20: Υλοποίηση των γεγονότων Action των κουμπιών των δέκα αριθμών

Βήμα 13: Η υλοποίηση του κουμπιού της τελείας μπορεί να γίνει με δύο τρόπους. Καταρχήν, την τελεία θα την γράψουμε στο *calcbox* όπως όλους τους αριθμούς. Φυσικά και στους δύο τρόπους θα πρέπει να εξασφαλίσουμε ότι η τελεία θα γράφεται μόνο μία φορά στον αριθμό.

Στην Εικόνα 21, παρουσιάζεται ο πρώτος τρόπος όπου χρησιμοποιείται μία σημαία για να ελέγχει κάθε φόρα αν περιέχει τελεία ο αριθμός που είναι στο *calcbox*. Έτσι, στη γραμμή 366, έξω από μεθόδους (και την main) μέσα στην τάξη, δηλώνουμε ως global τη Boolean μεταβλητή *dotflag*, η οποία θα έχει το ρόλο της σημαίας. Αρχική τιμή της μεταβλητής ορίζουμε την τιμή false που σημαίνει ότι δεν έχει μπει τελεία. Έτσι, στον κώδικα της ενέργειας ActionPerformed για την τελεία, ελέγχεται πρώτα η τιμή της dotflag. Αν είναι false, άρα δεν έχει μπει τελεία, τότε προσθέτουμε μία τελεία στο calcbox, δεξιά του υπάρχοντος αριθμού και αλλάζουμε την τιμή της dotflag σε true. Τέλος, κάθε φορά που γράφεται νέος αριθμός, αφού αδειάσει το calcbox, θα πρέπει η dotflag να γίνεται false.

323		
324	Ę	<pre>private void BdotActionPerformed(java.awt.event.ActionEvent evt) {</pre>
325		if (dotflag == false) {
326		<pre>calcbox.setText(calcbox.getText() + ".");</pre>
327		dotflag = true;
328		}
329	L	}
330		
331	+	/**3 lines */
334	+	<pre>public static void main(String args[]) {31 lines }</pre>
365		
366		<pre>boolean dotflag = false;</pre>
367		
368		// Variables declaration - do not modify
369		private javax.swing.JButton B0;
370		private javax.swing.JButton B1;
		<

Εικόνα 21: Υλοποίηση του γεγονότος Action του κουμπιού της τελείας – 1°ς τρόπος

Στην Εικόνα 22, παρουσιάζεται ο δεύτερος τρόπος όπου αφού γραφτεί η τελεία στο calcbox απενεργοποιείται το κουμπί της τελείας για να μην μπορεί ο Χρήστης να ξαναγράψει την τελεία. Φυσικά, όταν θα αδειάζει το *calcbox* θα πρέπει να ενεργοποιείται το κουμπί της τελείας.

323		
324	Ę	<pre>private void BdotActionPerformed(java.awt.event.ActionEvent evt) {</pre>
325		<pre>calcbox.setText(calcbox.getText() + ".");</pre>
326		<pre>Bdot.setEnabled(false);</pre>
327	L	}
328		



Βήμα 14: Το αριθμητήριο που παρουσιάζουμε είναι αρκετά απλό αφού θα διαχειρίζεται την πράξη δύο αριθμών. Στην Εικόνα 23, παρουσιάζεται ο κώδικας των τεσσάρων πράξεων. Έτσι, στα *Action* γεγονότα των κουμπιών των τεσσάρων πράξεων θα πρέπει να γίνουν τα παρακάτω:

- Να καταχωρηθεί ο πρώτος αριθμός της πράξης, ο αριθμός που είχε καταχωρηθεί στο calcbox πριν πατηθεί το πλήκτρο της πράξης σε μία double μεταβλητή – γραμμή 350. Η μεταβλητή number θα πρέπει να δηλωθεί ως global.
- Να ετοιμαστεί το calcbox να δεχτεί τον δεύτερο αριθμό. Με άλλα λόγια να αδειάσει το περιεχόμενό του – γραμμή 351.
- Το κουμπί της τελείας να αρχικοποιηθεί γραμμή 352. Σε περίπτωση που προτιμήθηκε η λύση με τη σημαία θα πρέπει να χρησιμοποιηθεί ο κώδικας στο σχόλιο της γραμμής 352.
- Να σημειωθεί σε μία global String μεταβλητή το σύμβολο της πράξης. Για παράδειγμα αν επιλεγεί η πράξη της πρόσθεσης θα σημειωθεί η πρόσθεση – γραμμή 353.



Εικόνα 23: Ο κώδικας των τεσσάρων πράξεων

Βήμα 15: Σε αυτό το βήμα θα προγραμματιστεί το ίσον. Έτσι, στην ενέργεια ActionPerformed του κουμπιού Bequal (Εικόνα 24) θα πρέπει να γίνουν τα παρακάτω:

- Γραμμή 390, αποθηκεύουμε σε μία String temp μεταβλητή το περιεχόμενο του calcbox, που λογικά θα περιέχει το δεύτερο αριθμό.
- Γραμμές 391 396, αν ο χρήστης πάτησε το ίσον πριν βάλει δεύτερο αριθμό τότε το αποτέλεσμα θα είναι ο πρώτος αριθμός, αν υπάρχει, αλλιώς το μηδέν.
- Αν ο χρήστης έβαλε δεύτερο αριθμό πριν πατήσει το ίσον, τότε στη γραμμή 398, μετατρέπεται σε double και αποθηκεύεται στη μεταβλητή number2.
- Γραμμή 399, αν η πράξη είναι πρόσθεση τότε γίνεται πρόσθεση.
- Γραμμή 400, αν η πράξη είναι αφαίρεση τότε γίνεται αφαίρεση.
- Γραμμή 401, αν η πράξη είναι πολλαπλασιασμός τότε γίνεται πολλαπλασιασμός.
- Γραμμές 402-408, αν η πράξη είναι διαίρεση τότε γίνεται διαίρεση και έλεγχος αν ο διαιρέτης είναι το μηδέν.
- Γραμμές 409-413, εμφανίζεται το αποτέλεσμα.

```
📑 Ergastirio2.java 🛛 🗙
Source Design History 🛛 🚱 🗸 💭 🗸 🖓 😓 🖓 😓 🖓 😓 😒 😒 😏 🕘 🔛
388
389
     private void BequalActionPerformed(java.awt.event.ActionEvent evt) {
390
               String temp = calcbox.getText();
391
               if (temp.isEmpty()) {
392
                   if (number1 != 0) {
                       calcbox.setText(String.valueOf(number1));
393
                   } else if(number1 == 0 ) {
394
395
                      calcbox.setText("0.0");
396
                   3
               } else {
397
                   number2 = Double.valueOf(temp);
398
399
                   if (sign.contentEquals("add")) result = number1 + number2;
                   else if(sign.contentEquals("minus")) result = number1 - number2;
400
                   else if(sign.contentEquals("times")) result = number1 * number2;
401
402
                   else if(sign.contentEquals("divide")) {
403
                        if (number2 != 0) {
                           result = number1 / number2;
404
405
                        } else {
                           errorflag = true;
406
407
408
                    3
                   if (errorflag == false) {
409
                       resulttext = String.valueOf(result);
410
411
                       calcbox.setText(resulttext);
412
                    } else calcbox.setText("ERROR");
413
               }
414
```

Εικόνα 24: Η ενέργεια ActionPerformed του ίσον (Bequal)

Βήμα 16: Για να αδειάσουμε το *textfield calcbox* θα χρησιμοποιήσουμε το κουμπί *C*. Με αυτήν την ενέργεια ξεκινάμε μια νέα πράξη. Οπότε είναι σημαντικό να αρχικοποιήσουμε όλες τις παραμέτρους του αριθμητήριου μας. Ο σχετικός κώδικας παρουσιάζεται στην Εικόνα 25, καθώς και η δήλωση όλων των global μεταβλητών (γραμμές 457-462).

👔 Ergastirio2.java 🛛 🗙			
Source	De	isign History 📴 😼 - 🐻 - 💐 🔩 🖓 🖓 🖓 🖓 🎧 🎧 🌾 🏀 🕮 🖆 👄 💷 些 🚅	
415			
416	F	<pre>private void BCActionPerformed(java.awt.event.ActionEvent evt) {</pre>	
417		<pre>calcbox.setText("");</pre>	
418		Bdot.setEnabled(true); // εναλλακτικά dotflag = false;	
419		errorflag = false;	
420		<pre>number1 = 0;</pre>	
421		<pre>number2 = 0;</pre>	
422		result = 0;	
423	L	}	
424			
425	+	<pre>public static void main(String args[]) {31 lines }</pre>	
456			
457		double number1 = 0;	
458		double number2 = 0;	
459		double result = 0;	
460		String sign = null;	
461		boolean errorflag = false;	
462		<pre>String resulttext = " ";</pre>	
463			

Εικόνα 25: Η ενέργεια ActionPerformed του κουμπιού που καθαρίζει το textfield (BC)

Βήμα 17: Τελευταίο βήμα είναι ο προγραμματισμός του <ΕΝΤΕR> να κάνει την ίδια ενέργεια με το ίσον (το κουμπί *Bequal*), όποτε το επιλέγει/πατάει ο χρήστης, ενώ ο δείκτης του ποντικιού είναι μέσα στο *textfield calcbox*. Στη γραμμή 10 της Εικόνα 26 παρουσιάζεται ο κώδικας που υλοποιεί το ζητούμενο. Το *defaultButton* είναι το <ΕΝΤΕR>, το οποίο ορίζεται, όποτε επιλέγεται, με την setDefaultButton μέθοδο να εκτελεί την ενέργεια που έχει προγραμματιστεί για το *Bequal*.

🔂 Ergastirio2.java 🛛 🗙		
Source	e Design History 📴 🛃 - 💭 - 💐 🖓 🖓 🖓 🖓 🖓 🥵 🖓 🖄 🖄 🎯 🗐 🖉 🚅	
1		
2	<pre>import java.awt.Component;</pre>	
3		
4	<pre>public class Ergastirio2 extends javax.swing.JFrame {</pre>	
5		
6	<pre>@SuppressWarnings("OverridableMethodCallInConstructor")</pre>	
7	public Ergastirio2() {	
8	<pre>initComponents();</pre>	
9	disable_components();	
10	<pre>getRootPane().setDefaultButton(Bequal);</pre>	
11		
12		

Εικόνα 26: Το <ΕΝΤΕR> λειτουργεί ως ίσον