## Εργαστήριο 4 - Άσκηση - Ανάλυση

**Εκφώνηση**: Δημιουργείστε την εφαρμογή «Audio Player» σε Java Swing με χρήση NetBeans ακολουθώντας τις παρακάτω οδηγίες.

- 1. Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame, ο οποίος θα έχει
  - τίτλο «Audio Player»,
  - διάταξη τύπου Grid Bag Layout.
- 2. Τοποθετήστε στο frame
  - 6 ετικέτες
  - 5 slider
  - 3 radio button
  - 1 button
- Επιλέξτε «Customize Layout» από το μενού του frame που ανοίγει με το δεξί πλήκτρο του ποντικιού. Τοποθετήστε τα συστατικά όπως φαίνεται παρακάτω:

(Σημείωση με την επιλογή grid size μπορείτε ένα συστατικό να το τοποθετείτε σε περισσότερα του ενός κελιά ενώ με το Anchor τα τοποθετείτε σχετικά στη στήλη, π.χ. αν επιλέξετε το Line Start θα είναι τοποθετημένα αριστερά οριζόντια).



4. Ονομάστε τα συστατικά text και variable name όπως φαίνεται στην παρακάτω εικόνα – για παράδειγμα για το πρώτο ορίστε για το jLabel το text: «Bass» και το variable name: «LBass» ενώ το variable name του jSlider «SlBass». Ονομάστε το variable name του τελευταίου label (jLabel6) «info». Τοποθετείστε το σε δύο κελιά και ορίστε μέγεθος του (preferredSize) 100, 150.

🕌 Audio Player		_	×
Bass	-10 -8 -6 -4 -2 0 2 4 6 8 10		
Midrange	-10 -8 -6 -4 -2 0 2 4 6 8 10		
Tremble	-10 -8 -6 -4 -2 0 2 4 6 8 10		
Balance	L Center R		
Volume	0 1 2 3 4 5 6 7 8 9 10		
● M0	○ M1 ○ M2		
	Bass: 0 Midrange: 0 Tremble: 0 Balance: Center Volume: 0		

- 5. Για το slider Bass Midrange, Tremble ορίστε τις παρακάτω ιδιότητες ως εξής:
  - 1. majorTickSpacing: 2
  - 2. minorTickSpacing: 1
  - 3. minimum: -10
  - 4. maximum: +10
  - 5. value: 0
  - 6. paintTicks: YES
  - 7. paintLabels: YES
- 6. Για το slider Volume ορίστε τις παρακάτω ιδιότητες ως εξής:
  - 1. majorTickSpacing: 1
  - 2. minimum: 0
  - 3. maximum: +10
  - 4. value: 0
  - 5. paintTicks: YES
  - 6. paintLabels: YES
- 7. Για το slider Balance ορίστε τις παρακάτω ιδιότητες ως εξής:
  - 1. majorTickSpacing: 2
  - 2. minorTickSpacing: 1

- 3. minimum: -5
- 4. maximum: +5
- 5. value: 0
- 6. paintTicks: YES
- 7. paintLabels: YES
- 8. Ανοίξτε το μενού του slider 4 (Balance) με το δεξί πλήκτρο του ποντικιού και επιλέξτε «customize Code» και προσθέστε στο κατάλληλο σημείο τον παρακάτω κώδικα:

table.put(new Integer(0), new JLabel("Center")); table.put(new Integer(-5), new JLabel("L")); table.put(new Integer(5), new JLabel("R")); SlBalance.setLabelTable(table);

Σημείωση: μην ξεχάσετε να δηλώσετε στις γενικές σας μεταβλητές τον πίνακα table ως

Hashtable table = new Hashtable();

Θα χρειαστεί να προσθέστε δύο βιβλιοθήκες – εκμεταλλευτείτε τις δυνατότητες του netbeans.

Τρέξτε το πρόγραμμα για να μελετήσετε τι κάνει το παραπάνω κομμάτι κώδικα.

- 9. Προσπαθήστε να κάνετε εύχρηστη την διεπαφή σας χρησιμοποιώντας για τα sliders το empty border. Πειραματιστείτε.
- 10. Δημιουργήστε μία βοηθητική τάξη Presets με τον κατάλληλο δομητή η οποία θα αποθηκεύει τιμές για τα πέντε sliders.
- 11. Δηλώστε ένα πίνακα Presets.
- 12. Δημιουργήστε μία μέθοδο setupPresets η οποία θα δίνει τις παρακάτω αρχικές τιμές στα πέντε slider ανά radiobutton:
  - a. M0 (0,0,0,0,0) Presets[0]
  - b. M1 (1, -1, 9, 0, 4) Presets[1]
  - c. M2 (2, 4, -2, 4, 2) Presets[2]

Μην ξεχάσετε να καλέσετε την μέθοδο με την εκκίνηση της εφαρμογής σας αμέσως μετά την initComponents().

- 13. Δημιουργήστε μία μέθοδο **showSliderValues** η οποία θα εμφανίζει τις τιμές των πέντε slider στην ετικέτα info. (Προτείνετε η χρήση της HTML)
- 14. Εμφανίστε στην Ετικέτα info την τρέχουσα κατάσταση των πέντε sliders όποτε συμβαίνει μία αλλαγή σε κάποιο συστατικό της διεπιφάνειας (Change - stateChanged).

- 15. Δημιουργήστε μία μέθοδο **loadPresets** φόρτωσης των αποθηκευμένων τιμών όποτε πατείτε το αντίστοιχο radio button (προγραμματίστε και το αντίστοιχο γεγονός).
- 16. Δημιουργήστε μία μέθοδο storePresets αποθήκευσης και προγραμματίστε το κουμπί JButton ώστε να αποθηκεύει τις τρέχουσες τιμές στα αντίστοιχα radiobutton.

Σημείωση: Μέθοδοι που πιθανώς θα σας φανούν χρήσιμες:

- JTextField.setText(string);
- JTextField.getText();
- Void JSlider.setValue(int)
- Int JSlider.getValue()

## <u>Ανάλυση και εξήγηση λύσης</u>

**Βήμα 1**: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi6.

**Βήμα 2**: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio4.

**Βήμα 3**: Ορισμός Ιδιοτήτων στο JFrame. Ορίζουμε ως τίτλο του JFrame το «Audio Player».

**Βήμα 4**: Ορίζουμε τη διάταξη σε Grid Bag Layout (Εικόνα 1).

Start Page >	🕻 📑 Ergastirio4.java 🛛 🗙		
Source Desi	gn History 🔣 🖶 🔮		#EE# +++
	Preview Design	>	
	Events	, ,	
	Set Layout	>	Free Design
	Customize Layout Add From Palette Design This Container Set to Default Size Edit Layout Space	2	Absolute Layout Border Layout (Default) Box Layout Card Layout Flow Layout
	Change Order		Grid Bag Layout
	<b>Copy</b> Paste	Ctrl+V	Grid Layout Null Layout Overlay Layout
	Customize Code Properties		



**Βήμα 5**: Τοποθετούμε στο JFrame έξι ετικέτες, πέντε slider, τρία radio button και ένα jButton, όπως φαίνεται στην Εικόνα 2.

Projects × Files	Services	-	Start Pag	e 🗙 📑	👌 Ergastii	io4.java 🚿	<								
🕀 🆢 Askisi 1			Source	Design	History	昭	2		1 I.I.	li 🕂	-44	▶ \$			
🖶 🆢 Askisi2															
😟 🍥 Askisi3															
🗄 🎒 Askisi4															
🗄 🍥 Askisi5															
🖨 🎒 Askisi6															
🖨 🚡 Source Packa	iges														
🖮 🧱 <default< td=""><td>package&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></default<>	package>														
Ergas	stirio4.java														
🗄 🍃 Libraries															
🗈 – 🎯 Lab3															
			il abel 1	i abel2il al	oel3iLabel4	il abel5il abelé		-	-	- <b>-</b> -	ι.	○ iRadioButton1	○ iRadioButton2	○ iRadioButton3	iButton1
			,		,	,,,			•	- C.	۰.	0,	0,	0,	
jButton1 [JButton] - Nav	vigator ×	-													
when iLabel4 [JLabe	ell	^													
wei jLabel5 [JLabe	el]														
wei jLabel6 [JLabe	el]														
jSlider 1 [JSlide	er]														
jSlider2 [JSlide	er]														
jSlider3 [JSlide	er]														
jSlider4 [JSlide	er]														
jSlider5 [JSlide	er]														
	1 [JRadioButton]														
	2 [JRadioButton]														
	3 [JRadioButton]														
JButton1 [JBu	itton]	~													

Εικόνα 2: Τοποθέτηση συστατικών στο JFrame

**Βήμα 6**: Με χρήση του εργαλείου Customize Layout (δεξί click στον καμβά για να επιλέξω το Customize Layout - Εικόνα 3) τοποθετώ τα συστατικά όπως μας ζητάει η εκφώνηση (Εικόνα 4).

Start Page X 🕞 Frgastirio4.java X							
Source Design History   □ 🛱 😁   🗮 🗮 🗮 🔚 👫 🗰 🔶 \$							
jLabel 1jLabel2jLabel3jLabel-	Preview Design Bind Events Set Layout Customize Layout Add From Palette Design This Container Set to Default Size Edit Layout Space Change Order Copy Paste Customize Code Properties	> > > > > > > > > > > > > > > > > > >	OjRadioButton1	○ jRadioButton2 ○ jRadioBut	ton3 jButton1		

**Εικόνα 3: Customize Layout** 

Start Pa	age ×	🖒 Ergastir	io4.java ×	:			
Source	Design	History	品		<u>∐i</u> # ⇔\$		
_				·			
				jLabel 1			
				jLabel2			
				jLabel3			
				jLabel4			
				jLabel5			
				) jRadioButton 1	◯ jRadioButton2	jRadioButton3	
					jButton 1	jLabel6	

Εικόνα 4: Η εμφάνιση των συστατικών μετά τη χρήση του Customize Layout

**Βήμα 7**: Αλλάζουμε text και variable names όπως αναφέρει η εκφώνηση και το αποτέλεσμα φαίνεται στην Εικόνα 5.

Source Design History   ↓ 2 🛱 🖤   🔤 🚍 III 🗽 🖶 🔶 🛊
Source besign History High I and I a

Εικόνα 5: Η εμφάνιση των συστατικών μετά την αλλαγή text των συστατικών

**Βήμα 8**: Για τα πέντε slider ορίζουμε εφτά ιδιότητες, όπως αναφέρονται στην εκφώνηση. Για παράδειγμα στην Εικόνα 6 φαίνεται ο ορισμός των ιδιοτήτων για το πρώτο slider. Οι ιδιότητες που ορίζουμε είναι: (α) majorTickSpacing, κάθε πόσες θέσεις θα εμφανίζεται η μεγάλη γραμμή κάτω από το slider, (β) minorTickSpacing, κάθε πόσες θέσεις θα εμφανίζεται η μικρή γραμμή κάτω από το slider, (γ) minimum, κάτω άκρο του slider, (δ) maximum, πάνω άκρο του slider, (ε) value, αρχική τιμή που θα εμφανίζεται ο δείκτης του slider, (στ) paintTicks, ορίζει αν θα εμφανιστούν οι γραμμές κάτω από το slider και (ζ) paintLabels, ορίζει αν θα εμφανιστούν οι

ετικέτες (αριθμοί) κάτω από το slider. Στην Εικόνα 7 παρουσιάζεται η εμφάνιση της εφαρμογής μετά τον ορισμό των παραπάνω ιδιοτήτων στα πέντε slider.

Start Pag	e 🛛 🔂 Ergast	🗴 🎧 SIBass [JSlider] - Properties		×					
Source	Design History	Properties Binding Events	Code						
		Properties		^					
		background	[240,240,240]						
		font	Tahoma 11 Plain						
		3a foreground	[240,240,240]						
		majorTick Spacing	2						
		ni maximum	10						
		minimum	-10						
		<sup>r</sup> minorTickSpacing	1						
	1	3a orientation	HORIZONTAL	~					
		paintLabels	$\checkmark$						
		paintTicks	$\checkmark$						
		paintTrack	$\checkmark$						
		snapToTicks							
		toolTipText							
		value	0						
		Other Properties							
		UIClassID	SliderUI						
		alignmentX	0.5						
		alignmentY	0.5						
		autoscrolls							
		baselineResizeBehavior	OTHER	$\sim$					
		border	(No Border)	v					
		SIBass [JSlider]	Close	(2) Help					

Εικόνα 6: Ορισμός ιδιοτήτων του SIBass



Εικόνα 7: Η εμφάνιση των συστατικών μετά τον ορισμό των ιδιοτήτων στα slider

**Βήμα 9**: Στο 4° slider, το slider balance θέλουμε οι ετικέτες να είναι διαφορετικές από το standard set ετικετών, δηλαδή τους αριθμούς. Θέλουμε στο αριστερό άκρο του slider να μπει η ετικέτα Left (στη θέση του -5), στο δεξί άκρο να μπει η ετικέτα Right (στη θέση του 5) και στο κέντρο το Center (στη θέση του 0).

Αυτό θα το υλοποιήσουμε με χρήση ενός Hashtable. Έτσι, δηλώνουμε ως global μεταβλητή ένα Hashtable, όπως βλέπουμε στην Εικόνα 8. Επίσης, πρέπει να προσθέσουμε μία βιβλιοθήκη για το Hashtable, την οποία μπορούμε να την προσθέσουμε με χρήση της επιλογής *Fix Imports*, του pop up μενού που ανοίγει με δεξί click πάνω στο Hashtable.

Τις ετικέτες θα τις ορίσουμε στο design, στον κώδικα υλοποίησης του slider. Έτσι, με δεξί click πάνω στο slider balance, επιλέγουμε από το pop up μενού που ανοίγει, την επιλογή *Customize Code*, όπως φαίνεται στην Εικόνα 9. Εκεί προσθέτουμε τον κατάλληλο κώδικα, όπως φαίνεται στην Εικόνα 10. Στις παρακάτω γραμμές κώδικα,

table.put(new Integer(0), new JLabel("Center")); table.put(new Integer(-5), new JLabel("L")); table.put(new Integer(5), new JLabel("R"));

στο Hashtable πίνακα *table*, ουσιαστικά ορίζουμε ότι στη θέση 0 του slider θα μπει η ετικέτα "Center", στη θέση -5 η ετικέτα "L" και στη θέση 5 η ετικέτα "R". Ενώ στη γραμμή κώδικα

SIBalance.setLabelTable(table);

Ορίζουμε ότι οι ετικέτες του slider SLBalance θα οριστούν με βάση το table.

Για να ολοκληρώσουμε αυτό το βήμα θα πρέπει να προσθέσουμε μία βιβλιοθήκη για χρήση της JLabel, με χρήση της γραμμής κώδικα import javax.swing.JLabel;

Οι νέες ετικέτες θα εμφανιστούν στην εκτέλεση του προγράμματος.

📩 Erg	🔂 Ergastirio4.java 🛛 🗙					
Source	Design	History 🛛 🕼 🖷 🗸 💐 🖓 😓 🕞				
215	-	<pre>});</pre>				
216	L	}				
217						
<b>S</b>		Hashtable table = new Hashtable()				
219		<pre>// Variables declaration - do</pre>	Navigate	>		
220		private javax.swing.JLabel LBa	Show Javadoc	Alt+F1		
221		private javax.swing.JLabel LBa	Find Usages	Alt+F7		
222		private javax.swing.JLabel LMi	Call Hierarchy			
223		private javax.swing.JLabel LTr		AB 1		
224		private javax.swing.JLabel LVo	Insert Code	Alt+Insert		
225		private javax.swing.JSlider Sl	Fix Imports	Ctrl+Shift+I		
226		private javax.swing.JSlider Sl	Refactor	>		
227		private javax.swing.JSlider Sl	Format	Alt+Shift+F		
228		private javax.swing.JSlider Sl	Run File	Shift+F6		
229		private javax.swing.JSlider Sl	Dahua Fila	Chill Childs FE		
230		private javax.swing.JLabel inf	Debug File	Ctri+Shirt+P5		
231		private javax.swing.JButton jB	lest File	Ctrl+F0		
232		private javax.swing.JRadioButt	Debug Test File	Ctrl+Shift+F6		
233		private javax.swing.JRadioButt	Run Focused Test Method			
234		private javax.swing.JRadioButt	Debug Focused Test Method			
235		<pre>// End of variables declaratio</pre>	Run Into Method			
236	}		New Watch	Ctrl+Shift+F7		
237			Toggle Line Breakpoint	Ctrl+F8		

Εικόνα 8: Δήλωση Hashtable και χρήση της Fix Imports



Εικόνα 9: Χρήση του Customize Code

Code Customizer		,
Component: SIBalance	∼ Rename	
nitialization code		
		^
default code	SlBalance = new javax.swing.JSlider();	
pre-init	<pre>table.put(new Integer(0), new JLabel("Center"));</pre>	
pre-init	<pre>v table.put(new Integer(-5), new JLabel("L"));</pre>	
pre-init	<pre>v table.put(new Integer(5), new JLabel("R"));</pre>	
pre-init	SlBalance.setLabelTable(table);	
default code	SlBalance.setMajorTickSpacing(2);	
default code	SlBalance.setMaximum(5);	
default code	SlBalance.setMinimum(-5);	
default code	SlBalance.setMinorTickSpacing(1);	
default code	SlBalance.setPaintLabels(true);	
default code	<pre>&gt; SlBalance.setPaintTicks(true);</pre>	
/ariable declaration	code	
		-
	<pre>private javax.swing.JSlider SlBalance;</pre>	
ariable: field	Access: private V final static transient volatile	
		1 11-1-

Εικόνα 10: Ο κώδικας του Customize Code στο slider balance

**Βήμα 10**: Θα βελτιώσουμε τώρα την εμφάνιση της εφαρμογής μας. Αυτό θα το πετύχουμε απομακρύνοντας τα slider για να γίνουν πιο ευδιάκριτα με χρήση των empty borders. Έτσι, επιλέγουμε για κάθε slider την ιδιότητα border και επιλέγουμε το τύπο empty border. Ορίζουμε τις τιμές 20, 10, 10 και 10, όπως φαίνεται στην Εικόνα 11.

Επίσης, ορίζουμε preferredSize και minimumSize του jFrame τις τιμές 550 και 600 για να μην «συμπιεστούν» τα συστατικά από ένα μικρό σε μέγεθος Frame. Στην Εικόνα 12, βλέπουμε την εμφάνιση του design μετά την εφαρμογή των παραπάνω.

🕽 Ergastirio4.java 🛛 🛛			Palette ×
Source Design History 🛄 🖶 😁	22##EH ++	SIBass [JSlider] - border	×
		Set SiBass's border property using: Border customizer	~
Bass Midrange Tremble Balance Volume () RadioButton	-10-8-6-4-2 0 2 4 6 8 10 -10-8-6-4-2 0 2 4 6 8 10 -10-8-6-4-2 0 2 4 6 8 10 -10-8-6-4-2 0 2 4 6 8 10 -5-3-1 1 3 5 0 1 2 3 4 5 6 7 8 9 10 1 0 RadoButton2 ()RadoButton3 [putton1] info	Available Borders  (%o Border)  Bevel Border  Compound Border  Etched Border  Etched Border  Soft Bevel Border  Title Border  Title Border  Totel Border  Insets  [20, 10, 10, 10]	
		Empty Border javax.swing.border.EmptyBorder An empty and transparent border which takes up space but does no drawing	
		CK Cancel	Help

Εικόνα 11: Χρήση του empty border για τα πέντε slider

📑 Ergas	tirio4.java ×	
Source	Design History	▝▏▙▋፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟፟
	Bass	-10-8 -6 -4 -2 0 2 4 6 8 10
	Midrange	-10-8 -6 -4 -2 0 2 4 6 8 10
	Tremble	-10-8 -6 -4 -2 0 2 4 6 8 10
	Balance	-5 -3 -1 1 3 5
	Volume	0 1 2 3 4 5 6 7 8 9 10
	◯ jRadioButton1	◯ jRadioButton2 ◯ jRadioButton3
		jButton1 info

Εικόνα 12: Η εμφάνιση του design μετά το βήμα 10

**Βήμα 11**: Για να διαχειριστούμε τα radiobuttons ως μνήμες στο audioplayer, θα πρέπει να αποθηκεύουμε τρία σετ τιμών των sliders σε μία δομή. Αυτό θα το κάνουμε με τη δημιουργία της τάξης Presets. Έτσι, θα δημιουργήσουμε ένα αρχείο Java Class όπου εκεί θα ορίσουμε μία κλάση Presets με πέντε ακέραιες ιδιότητες, μία για κάθε δείκτη των πέντε slider – ένας δείκτης στα slider δείχνει μία ακέραια τιμή. Στην Εικόνα 13 φαίνεται η κλάση Presets. Επίσης, δηλώνουμε ένα πίνακα αντικειμένων Presets με διάσταση τρία, όσα είναι τα radiobuttons με την εντολή Presets pinakas[] = new Presets[3];

📑 Ei	rgastirio4.java 🗙 🗟 Presets.java 🗙
Sour	ce History   🞯 💀 - 🔊 -   🔍 🖓 🖓 🚭 📪   🄗 🈓   🗐 🗐 🥥
1	<pre>public class Presets {</pre>
2	int bass;
3	<pre>int midrange;</pre>
4	<pre>int tremble;</pre>
5	<pre>int balance;</pre>
6	<pre>int volume;</pre>
7	
8	Presets(int a, int b, int c, int d, int e) {
9	bass = a;
10	midrange = b;
11	<pre>tremble = c;</pre>
12	balance= d;
13	volume = e;
14	L }
15	}

Εικόνα 13: Η τάξη Presets

**Βήμα 12**: Σε αυτό το βήμα θα δώσουμε αρχικές τιμές στα τρία αντικείμενα του πίνακα. Πριν το κάνουμε αυτό θα διαμορφώσουμε την εμφάνιση των τριών radiobutton. Το κείμενο των τριών radiobuttons θα γίνει M0, M1 και M2 και οι μεταβλητές τους RBM0, RBM1, και RBM2 αντίστοιχα. Φροντίζουμε να «μπουν» και τα τρία στο ίδιο buttongroup και στο RBM0 να οριστεί η ιδιότητα selected ώστε να εμφανίζεται επιλεγμένο το πρώτο radiobutton όταν ξεκινά η εφαρμογή. Επίσης, αλλάζουμε, από το *customize layout* το insets των στοιχείων (radiobutton και jbutton) για καλύτερη στοίχιση και εμφάνιση.

Για την απόδοση αρχικών τιμών στα τρία αντικείμενα του πίνακα δημιουργούμε τη μέθοδο setupPresets - Εικόνα 14, γραμμές κώδικα 236 - 240. Την μέθοδο την καλούμε από το δομητή της τάξης μετά την initComponents() για να αποδοθούν οι αρχικές τιμές στην εκκίνηση της εφαρμογής.

🗊 Ergastirio4.java 🗙 🔊 Presets.java 🗙							
Source Design History 🕼 💀 - 🗟 - 🕄 - 🔍 🖓 😓 🖓 🔮 🆓 😓 🖄							
1	Ę	<pre>import java.util.Hashtable;</pre>					
2	L	<pre>import javax.swing.JLabel;</pre>					
3							
4	<pre>4 public class Ergastirio4 extends javax.swing.JFrame {</pre>						
5							
6	F	<pre>public Ergastirio4() {</pre>					
7		<pre>initComponents();</pre>					
8		<pre>setupPresets();</pre>					
9	L	}					
10							
11		<pre>@SuppressWarnings("unchecked")</pre>					
12	+	Generated Code					
203							
204	+	<pre>public static void main(String args[]) {31 lines }</pre>					
235							
236	F	<pre>void setupPresets() {</pre>					
237		pinakas[0] = new Presets(0,0,0,0,0);					
238		pinakas[1] = new Presets(1,-1,9,0,4);					
239		pinakas[2] = new Presets(2,4,-2,4,2);					
240	L	}					
241							
242		Hashtable table = new Hashtable();					
243		<pre>Presets pinakas[] = new Presets[3];</pre>					
244							
245		// Variables declaration - do not modify					
246		private javax.swing.JLabel LBalance;					
247		private javax.swing.JLabel LBass;					
248		<pre>private javax.swing.JLabel LMidrange;</pre>					

Εικόνα 14: Η μέθοδος setupPresets

**Βήμα 13**: Το επόμενο που θα υλοποιήσουμε είναι η εμφάνιση της θέσης των δεικτών των slider στην ετικέτα info. Για να εμφανίσουμε τις τιμές των πέντε δεικτών σε πέντε σειρές θα χρησιμοποιήσουμε html. Επίσης, τη θέση του δείκτη του slider balance θα την παρουσιάσουμε σε συνάρτηση με τις ετικέτες Left, Center και Right. Στην Εικόνα 15, φαίνεται ο κώδικας υλοποίησης της μεθόδου showSliderValues. Με τη μέθοδο getValue (π.χ. γραμμή κώδικα 239) παίρνουμε την ακέραια τιμή που δείχνει ο δείκτης του slider που δείχνει. Για να γράψουμε html σε μία ετικέτα, θα πρέπει στη setText να ξεκινήσουμε το κείμενο με <html> (γραμμή κώδικα 247). Για να αλλάξουμε σειρά χρησιμοποιούμε το tag της html το <br> (π.χ. γραμμή κώδικα 248). Τέλος, την μέθοδο showSliderValues την καλούμε από το δομητή της τάξης μετά την initComponents() (π.χ. γραμμή κώδικα 9) για να εμφανιστούν οι αρχικές τιμές στην ετικέτα info στην εκκίνηση της εφαρμογής.

Ergastirio4.java × 🔊 Presets.java ×				
Source	e D	esign History   🕼 🛃 = 💐 = 💐 🖓 🖓 🖓 🖓 🔛 📪 🖓 🗞 🏷 🦉 🗐 🗐 🥥 🗐 🖉 🚅		
6	Ę	<pre>public Ergastirio4() {</pre>		
7		<pre>initComponents();</pre>		
8		<pre>setupPresets();</pre>		
9		<pre>showSliderValues();</pre>		
10	L	}		
11				
12		<pre>@SuppressWarnings("unchecked")</pre>		
13	+	Generated Code		
204				
205	+	<pre>public static void main(String args[]) {31 lines }</pre>		
236				
237	P	<pre>void showSliderValues() {</pre>		
238		String bal;		
239		<pre>int b = SlBalance.getValue();</pre>		
240		if(b > 0)		
241		<pre>bal = "Right " + String.valueOf(b);</pre>		
242		else if(b==0)		
243		<pre>bal = "Center";</pre>		
244		else		
245		<pre>bal = "Left " + String.valueOf(-b);</pre>		
246				
247		<pre>info.setText("<html> Bass: " + SlBass.getValue() +</html></pre>		
248		<pre>" Midrange: " + SlMidrange.getValue() +</pre>		
249		<pre>" Tremble: " + SlTremble.getValue() +</pre>		
250		" Balance: " + bal +		
251		<pre>" Volume: " + SlVolume.getValue());</pre>		
252		}		
253				

Εικόνα 15: Η μέθοδος showSliderValues

**Βήμα 14**: Στόχος της εφαρμογής είναι να εμφανίζει άμεσα την τρέχουσα τιμή κάθε δείκτη των slider. Αυτό θα το υλοποιήσουμε με τη δημιουργία του γεγονότος State-Changed, όπως φαίνεται στην Εικόνα 16. Στην ενέργεια κάθε γεγονότος θα γίνει κλήση της showSliderValues.





🖹 Ergastirio4.java 🗙 🙆 Presets.java 🗙				
Source	D	besign History   📴 🗔 - 👼 - 💐 - 💐 🚭 🖓 🖶 🎧 🖗 😓 🖄 🗐 🗐 🥥 🗐 🖉 🚅		
6	Ę	<pre>public Ergastirio4() {</pre>		
7		<pre>initComponents();</pre>		
8		<pre>setupPresets();</pre>		
9		<pre>showSliderValues();</pre>		
10	L	}		
11				
12		<pre>@SuppressWarnings("unchecked")</pre>		
13	+	Generated Code		
229				
230	Ę	<pre>private void SlBassStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
231		<pre>showSliderValues();</pre>		
232	L	}		
233				
234	Ę	<pre>private void SlMidrangeStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
235		<pre>showSliderValues();</pre>		
236	L	}		
237				
238	F	<pre>private void SlTrembleStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
239		<pre>showSliderValues();</pre>		
240	L	}		
241				
242	F	<pre>private void SlBalanceStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
243		<pre>showSliderValues();</pre>		
244	L	}		
245				
246	F	<pre>private void SlVolumeStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
247		<pre>showSliderValues();</pre>		
248	L	}		
0.40				

## Εικόνα 17: Ο κώδικας των γεγονότων State-Changed των sliders

**Βήμα 15**: Σε αυτό το βήμα θα υλοποιήσουμε τη χρήση των τριών radiobutton μνημών. Έτσι, κάθε φορά που θα πατάμε ένα radiobutton θα πρέπει να αλλάζουν οι θέσεις των δεικτών των slider σύμφωνα με τις αποθηκευμένες τιμές στο αντίστοιχο αντικείμενο του πίνακα της τάξης Presets. Ταυτόχρονα θα ενημερώνουμε κατάλληλα την εμφάνιση των τιμών των δεικτών στην

ετικέτα info, καλώντας τη μέθοδο showSliderValues. Αυτά θα τα υλοποιήσουμε δημιουργώντας τη μέθοδο loadPresets (γραμμές κώδικα 309-316). Με τη μέθοδο setValue() τοποθετούμε το δείκτη ενός slider στη θέση που δείχνει ο ακέραιος αριθμός που περιέχει η παράμετρος της μεθόδου (π.χ. data.bass). Η κλήση της loadPresets, γίνεται από το κατάλληλο γεγονός itemStateChanged των τριών radioButtons (γραμμές κώδικα 265-275).

👔 Ergastirio4.java 🗙 🙆 Presets.java 🗙				
Source	D	esign History 🛛 🚱 🕶 💭 🚽 💐 😓 😓 😓 🖓 😓 😓 🖓 😓 🔛		
200		1		
260				
261	P	<pre>private void SlVolumeStateChanged(javax.swing.event.ChangeEvent evt) {</pre>		
262		<pre>showSliderValues();</pre>		
263	L	}		
264				
265	무	<pre>private void RBMOItemStateChanged(java.awt.event.ItemEvent evt) {</pre>		
266		<pre>loadPresets(pinakas[0]);</pre>		
267	L	}		
268				
269	Ę	<pre>private void RBM1ItemStateChanged(java.awt.event.ItemEvent evt) {</pre>		
270		<pre>loadPresets(pinakas[1]);</pre>		
271	L	}		
272				
273	Ģ	<pre>private void RBM2ItemStateChanged(java.awt.event.ItemEvent evt) {</pre>		
274		<pre>loadPresets(pinakas[2]);</pre>		
275	L	}		
276				
277	+	<pre>public static void main(String args[]) {31 lines }</pre>		
308				
309	Ę	<pre>void loadPresets(Presets data) {</pre>		
310		SlBass.setValue(data.bass);		
311		<pre>SlMidrange.setValue(data.midrange);</pre>		
312		SlTremble.setValue(data.tremble);		
313		SlBalance.setValue(data.balance);		
314		SlVolume.setValue(data.volume);		
315		<pre>showSliderValues();</pre>		
316	L	}		
317				

Εικόνα 18: Η μέθοδος loadPresets και τα γεγονότα των radiobutton κλήσης της

**Βήμα 16**: Τελευταίο βήμα για την ολοκλήρωση της εφαρμογής είναι ο προγραμματισμός της αλλαγής των τιμών μίας μνήμης. Για την αποθήκευση των νέων τιμών των δεικτών των slider στο κατάλληλο αντικείμενο του πίνακα χρησιμοποιείται η μέθοδος storePresets (γραμμές κώδικα 323-329). Η κλήση της storePresets, γίνεται από το γεγονός ActionPerformed του Store jButton (γραμμές κώδικα 282-289) με παράμετρο το αντικείμενο του πίνακα Presets που αντιστοιχεί στο radiobutton που είναι επιλεγμένο την στιγμή που πατήθηκε το Store jButton.

🗟 Erga	astirio	4.java	X 🔊 Presets.java X
Source	De	esign	History   📴 😼 - 🐺 - 💐 🕄 😓 🖓 😓 🎧 🖓 😓 🖄 🖄 ڬ ڬ 🔛 🖉 🚢
281			
282	F		<pre>private void StoreButtonActionPerformed(java.awt.event.ActionEvent evt) {</pre>
283			<pre>if (RBM0.isSelected())</pre>
284			<pre>storePresets(pinakas[0]);</pre>
285			<pre>else if (RBM1.isSelected())</pre>
286			<pre>storePresets(pinakas[1]);</pre>
287			<pre>else if (RBM2.isSelected())</pre>
288			<pre>storePresets(pinakas[2]);</pre>
289	L		}
290			
291	+		<pre>public static void main(String args[]) {31 lines }</pre>
322			
323	F		<pre>void storePresets(Presets data) {</pre>
324			<pre>data.bass = SlBass.getValue();</pre>
325			<pre>data.midrange = SlMidrange.getValue();</pre>
326			<pre>data.tremble = SlTremble.getValue();</pre>
327			<pre>data.balance = SlBalance.getValue();</pre>
328			<pre>data.volume = SlVolume.getValue();</pre>
329	L		}
330			

Εικόνα 19: Η μέθοδος storePresets και το γεγονός του button store

.