Εργαστήριο 3 - Άσκηση - Ανάλυση

Εκφώνηση: Δημιουργείστε την εφαρμογή «Αγορά Σπιτιού» σε Java Swing με χρήση NetBeans ακολουθώντας τις παρακάτω οδηγίες.

- Η εφαρμογή θα σχεδιασθεί σε ένα εξωτερικό υποδοχέα JFrame, ο οποίος θα έχει τίτλο «Αγορά Σπιτιού» και θα οριστεί με την έξοδο του να κλείνει και η εφαρμογή.
 - Στο JFrame θα τοποθετήσετε ένα JPanel με ιδιότητες:
 - Background: επιλογής σας προσέξτε την αντίθεση με το background του JFrame
 - Border: TitledBorder με τίτλο «Στοιχεία Δανείου» με font επιλογής σας.
 - Επίσης, στο JPanel θα τοποθετήσετε τρία JLabel και τρία JTextField όπως φαίνονται στην εικόνα.
 - Στο JFrame θα τοποθετήσετε ένα δεύτερο JPanel με ιδιότητες:
 - Background: επιλογής σας προσέξτε την αντίθεση με το background του JFrame
 - Border: TitledBorder με τίτλο «Διαθέσιμα Σπίτια» με font επιλογής σας.
 - Επίσης, στο JPanel θα τοποθετήσετε τρία radiobuttons (μην ξεχάσετε να ορίσετε το ButtonGroup τους) και δύο Labels:
 - Το πρώτο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα χωρίς κείμενο. Στην ιδιότητα icon
 θα ορίσουμε το αρχείο εικόνας ChatsworthHouse.jpg.
 - Όταν πατιέται (pressedIcon) ή επιλέγεται (selectedIcon) ή φωτίζεται προσωρινά (rolloverIcon) η εικόνα θα αλλάζει σε (ChatsworthHousePressed.jpg).
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Να είναι αρχικά επιλεγμένη (ιδιότητα selected).
 - Το δεύτερο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα (queen_house.jpg)
 - Όταν πατιέται ή επιλέγεται ή φωτίζεται προσωρινά η εικόνα θα αλλάζει σε (queen_housePressed.jpg)
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Το τρίτο RadioButton θα έχει τις παρακάτω ιδιότητες:
 - Θα περιέχει μόνο εικόνα (white_house.jpg)
 - Όταν πατιέται ή επιλέγεται ή φωτίζεται προσωρινά η εικόνα θα αλλάζει σε (white_housePressed.jpg)
 - Border EtchedBorder με πλαίσιο (BorderPainted)
 - Δύο Labels όπως φαίνεται στην εικόνα.
- Επίσης, στο JFrame θα τοποθετήσετε ένα button και δύο Label όπως φαίνεται στην εικόνα.

	d	Αγορά σπιτιού	 ×
Στοιχεία Δανείου Κεφαλαίο: Επιτόκιο: Διάρκεια:	0 0 0 μήνες	- Διαθέσιμα Σπίτια	
Υπολογισμός Δόσης Δόση ανά Μήνα: Ο			
		Atia: 300000	

Στη συνέχεια θα προγραμματίσετε τη συμπεριφορά των συστατικών της διεπιφάνειας σας. Αναλυτικά:

- Ορίστε στα radiobutton το γεγονός **Item**StateChanged ώστε όταν επιλέγεται το κουμπί να αλλάζει η εικόνα του κουμπιού στη μορφή "…pressed" και να ορίζεται τιμή του σπιτιού σε
 - i. 3.000.000 για το 1° σπίτι,
 - іі. 5.000.000 у
ια το 2° σπίτι και
 - iii. 4.000.000 για το 3° σπίτι.
- Ορίστε στα radiobutton τα γεγονότα MouseEntered και MouseExited ώστε όταν μπαίνει ο δείκτης του ποντικιού να αλλάζει προσωρινά η αντίστοιχη τιμή στην ετικέτα, ενώ όταν βγαίνει να επιστρέφει στην προηγούμενη κατάσταση.
- Ορίστε στο κουμπί «Υπολογισμός Δόσης» το γεγονός ActionPerformed ώστε να υπολογίζει την κατάλληλη δόση ανάλογα με τις τιμές που έχουν τα συστατικά των δύο πάνελ. Το αποτέλεσμα να το εμφανίζει στην κατάλληλη ετικέτα.
- 4. Ορίστε συντομεύσεις για τα παρακάτω συστατικά:
 - i. Κεφάλαιο το αγγλικό κουμπί 'Κ',
 - ii. Επιτόκιο το αγγλικό κουμπί 'Ε',
 - iii. Διάρκεια το αγγλικό κουμπί 'D'
 - iv. Υπολογισμός Δόσης το αγγλικό κουμπί 'Υ',
 - v. 1o radiobutton τον αριθμό 1,
 - vi. 20 radiobutton τον αριθμό 2,
 - vii. 3o radiobutton τον αριθμό 3.

Μη ξεχάσετε να κάνετε τις απαραίτητες ενέργειες για να φαίνεται η κατάλληλη υπογράμμιση στα 4 πρώτα συστατικά που θα οριστεί συντόμευση ώστε να ενημερώνεται ο χρήστης.

Σημείωση: Μέθοδοι που πιθανώς θα σας φανούν χρήσιμες:

- JTextField.setText(string);
- JTextField.getText();
- String.valueOf(double);
- Double.valueOf(string);
- Integer.valueOf(string);
- Πεδίo_String.contentEquals(string)

Ανάλυση και εξήγηση λύσης

Βήμα 1: Δημιουργούμε ένα νέο project στο NetBeans με όνομα Askisi5.

Βήμα 2: Δημιουργούμε ένα νέο Java αρχείο τύπου JFrame Form με όνομα Ergastirio3.

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

Βήμα 4: Τοποθετούμε στον καμβά ένα JPanel. Το JPanel είναι ένα Swing Container, και το βρίσκουμε στην εργαλειοθήκη της Java Swing στα Swing Container. Από τις ιδιότητες του JPanel αλλάζουμε το χρώμα του background – επιλέξτε ένα απαλό χρώμα που όμως να είναι διακριτό από το γκρι χρώμα του JFrame. Στη συνέχεια αλλάζουμε το περίγραμμα (border) του JPanel σε titled border. Από τις ιδιότητες του JPanel επιλέγουμε τις «…» δεξιά στην ιδιότητα *border*. Από το παράθυρο που θα ανοίξει από τα «Available Borders» επιλέγουμε το «Titled Border». Στις ιδιότητες (Properties) του «Titled Border» (στο κάτω μέρος του παραθύρου) ορίζουμε τις προτιμήσεις μας. Έτσι, στο «Title» ορίζουμε ως τίτλο του JPanel «Στοιχεία Δανείου».

Start Page X R Ergacticio 2 java X R	jPanel1 [JPanel] - border	×	٦
Source Design History	Set jPanel1 's border property using:	Border customizer V	
	Available Borders (No Border) Bevel Border Compound Border Etched Border Line Border Matte Border Soft Bevel Border Titled Border		
	Properties Border Title Color Font Justification Position	[XPImageBorder] Στοιχεία Δανείου ■ [0,0,0] Tahoma 11 Plain Default Justification ~ Default Position ~	
Output ×	Titled Border javax.swing.border.TitledBorder A border with a String title in a speci	fied position and justification.	

Εικόνα 1: Ορισμός Titled Border

Βήμα 5: Τοποθετούμε τέσσερα JLabel και τρία TextField στο JPanel. Ορίζουμε τα ονόματα μεταβλητών και το κείμενο τους όπως φαίνεται στην Εικόνα 2. Επίσης, στα τρία JTextField ορίζουμε στην ιδιότητα *HorizontalAlignement* την τιμή *RIGHT* για δεξιά στοίχιση του περιεχομένου JTextField.





Βήμα 6: Τοποθετούμε στον καμβά ένα δεύτερο JPanel. Από τις ιδιότητες του JPanel αλλάζουμε το χρώμα του background και το περίγραμμα (border) του JPanel σε titled border. Στο «Title» ορίζουμε ως τίτλο του JPanel «Διαθέσιμα Σπίτια». Επίσης, προσθέτουμε στο JPanel τρία jRadioButtons, δύο JLabels και ένα ButtonGroup. Και τα τρία συστατικά τα βρίσκουμε στα Swing Controls. Το buttonGroup είναι αόρατο στον καμβά. Στο navigator τοποθετείται στα «*other components»*, εκεί όπου τοποθετούνται όλα τα αόρατα συστατικά. Το buttongroup θα το χρησιμοποιήσουμε για να ομαδοποιήσουμε σε μία ομάδα τα τρία jRadioButtons – θα το δούμε στο επόμενο βήμα.





Βήμα 7: Τα RadioButtons έχουν την ιδιαιτερότητα, όταν οριστούν να ανήκουν στην ίδια ομάδα, να είναι επιλεγμένο μόνο ένα RadioButton από όλη την ομάδα. Με το ButtonGroup δηλώνουμε ομάδες RadioButton. Στο προηγούμενο βήμα προσθέτοντας το ButtonGroup δημιουργήθηκε η ομάδα με όνομα buttonGroup1. Για να ορίσουμε τα τρία RadioButtons να είναι στην ίδια ομάδα, ορίζουμε να έχουν και τα τρία για την ιδιότητα buttonGroup την τιμή buttonGroup1, όπως φαίνεται στην Εικόνα 4.

👌 Erga	stirio3.jav	a X					4	Palett	e ×	-
Source	Design	History	12 🕄 📑 😭		E E + ⇔ 9	;		🙀 🗆 Sw	ing Containers	<u>۸</u>
				1	· Mill man a 1 · ·			Pa	nel	Tabbed Pane
						House_1RB [JRadioButton]	Properties	×	Pane	Scroll Pane
	Στοιχεία Δ	ανείου		ΓΔισθ	θέσιμα Σπίπια	Properties Binding Events	Code		Bar	🔁 Desktop Pane
	κεφαλαίο:		0		9	Properties			nal Frame	Layered Pane
	Επιτόκιο:		0		jRadioButton1	action	<none></none>	v	g Controls	
					1	background	[240,240,240]		1	OK Button
	Διάρκεια:		0 μήνες			model	<default></default>	v	le Button	R- Check Box
					- jRadioButton2	buttonGroup	buttonGroup1	v	ic botton	
				•		font	<none></none>		o Button	S- Button Group
						foreground	buttonGroup1		po Box	Eist List
					- jRadioButton3	mnemonic			Field	t× Text Area
						selected			Bar	D Slider
						text	jRadioButton 1			Competend Field
					Aller disheld	toolTipText			ress bar	Pormatted Field
					Acjo: jLabeiz	Other Properties			1RB [JRadioE	Sutton] - Prope × –
						UIClassID	RadioButtonUI		ies Binding	Events Code
						actionCommand	jRadioButton 1		ties	A
						alignmentX	0.0			<none></none>
						alignmentY	0.5		ind	
						autoscrolls				<default></default>
						baselineResizeBehavior	CENTER_OFFSET		roup	buttonGroup1 🗸
						border	CompoundBorderUlResou	rcej		Tahoma 11 Plain
						borderPainted			nd	[0.0.0]
						componentPopupMenu	<none></none>	~	c	
						contentArearned			-	
						buttonGroup		0		jRadioButton1
						Group of buttons in which this butto	n belongs		×t	🗸
							an belongs		IDP [IDedieP	utten]
									TKD [JRACIOD	uttonj 😈
							Cl	ose Help		
								p		

Εικόνα 4: Ομαδοποίηση των τριών jRadioButton

Βήμα 8: Τα radioButton δεν θα έχουν περιεχόμενο κείμενο αλλά εικόνα. Έτσι, θα σβήσουμε το κείμενο από τα radioButton, δηλαδή από την ιδιότητα text θα σβήσουμε το jRadioButton1 για το πρώτο κουμπί. Στη συνέχεια θα ορίσουμε στην ιδιότητα *icon* την εικόνα του αρχείου «ChatsworthHouse.jpg».

Τα έξι αρχεία εικόνων μπορείτε να τα αποθηκεύσετε σε ένα προσωρινό κατάλογο. Τα αρχεία εικόνων θα τα εισάγουμε στο project καθώς θα ορίζουμε τις ιδιότητες. Έτσι, για να ορίσουμε στην ιδιότητα *icon* το κατάλληλο αρχείο εικόνας αλλά και να εισάγουμε την εικόνα στο project, θα επιλέξουμε από τις ιδιότητες του πρώτου radioButton τις «…» δεξιά της ιδιότητας icon. Στο παράθυρο που θα ανοίξει θα επιλέξουμε το κουμπί «Import To Project». Στη συνέχεια θα βρούμε το αρχείο εικόνας «ChatsworthHouse.jpg», θα το επιλέξουμε και θα επιλέξουμε το κουμπί «Next» για να πάμε στο επόμενο βήμα. Μπορούμε την εικόνα να την εισάγουμε στο src κατάλογο του project ή να δημιουργήσουμε ένα νέο κατάλογο, όπως φαίνεται στην Εικόνα 5, και επιλέγουμε «Finish». Κλείνουμε το παράθυρο και παρατηρούμε (Εικόνα 6) ότι το radioButton περιέχει την εικόνα αλλά και την εμφάνιση του καταλόγου icons ως package στο project.

🗊 Import images to project	×
Steps	Select target folder
 Select image files to import Select target folder 	Classpath ⊟-↓ Source Packages ↓ ↓ cone
	Folder Name: Icons Create New
	< Back Next > Finish Cancel Help

Εικόνα 5: Δημιουργία του καταλόγου icons



Εικόνα 6: Το jRadioButton περιέχει εικόνα

Στη συνέχεια ορίζουμε στην ιδιότητα **pressedicon** την εικόνα του αρχείου εικόνας «ChatsworthHousePressed.jpg». Η συγκεκριμένη ιδιότητα ορίζει, ότι όταν θα επιλέγει ο χρήστης το radiobutton με το αριστερό πλήκτρο του ποντικιού θα αλλάζει η εικόνα του κουμπιού σε αυτή που περιέχει το αρχείο εικόνας. Την ίδια εικόνα ορίζουμε και στις ιδιότητες **rollovericon**, η εικόνα του κουμπιού θα αλλάζει στην εικόνα του αρχείου εικόνας που ορίσαμε στην ιδιότητα όποτε περνάμε το ποντίκι μέσα από το κουμπί χωρίς να πατήσουμε κάποιο πλήκτρο του ποντικιού, και **selectedicon**, η οποία ορίζει ποια εικόνα θα εμφανίζεται στο κουμπί όταν αυτό θα είναι το επιλεγμένο της ομάδας – ως επιλεγμένο ορίζεται αυτό που έχει δηλωθεί αρχικά να έχει την ιδιότητα selected (Εικόνα 7) ή το τελευταίο που θα επιλέξουμε με το αριστερό πλήκτρο του ποντικιού.

	🜍 House_1RB [JRadioButton] - P	roperties	×
	Properties Binding Events	Code	
	Properties		^
- Διαθέσιμα Σπίτια	action	<none></none>	~
	background	[240,240,240]	
<u> </u>	model	<default></default>	~
The second second	buttonGroup	buttonGroup1	~
The Production	font	Tahoma 11 Plain	
	foreground	[0,0,0]	
	mnemonic		
	selected		
••	text	_	
- jRadioButton2	toolTipText	True	
0.	Other Properties		
	UIClassID	RadioButtonUI	
O iRadioButton3	actionCommand		
ASia	alignmentX	0.0	
- ngu.	alignmentY	0.5	
	autoscrolls		
	baselineResizeBehavior	CENTER_OFFSET	
	border	[CompoundBorderUIResource]	
	borderPainted		
	componentPopupMenu	<none></none>	~
	contentAreaFilled		
		le C li e	
	selected		
	(boolean) selected		
		Close	Help

Εικόνα 7: Ιδιότητα selected

Τέλος, ορίζουμε το radiobutton να έχει την κατάλληλη τιμή στην ιδιότητα border και στην ιδιότητα borderPainted, όπως φαίνεται στην Εικόνα 8.

	House 1RB [JRadioButton]	- Properties	X
∃₽ĽĽ⊬∣⇔ \$	Dreparties Diadian Event	n Cada	
	Properdes binding Evend	is code	
	- Properties		^
Διαθέσιμα Σπίπια	action	<none></none>	×
	background	[240,240,240]	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	model	<default></default>	~
ALL DISTORT	buttonGroup	buttonGroup1	×
	font	Tahoma 11 Plain	
States	foreground	[0,0,0]	
	mnemonic		
· · · ·	selected	\checkmark	
••	text		
- jRadioButton2	toolTipText		
	Other Properties		
	UIClassID	RadioButtonUI	
jRadioButton3	actionCommand		
AEio:	alignmentX	0.0	
	alignmentY	0.5	
	autoscrolls		
	baselineResizeBehavior	CENTER_OFFSET	\sim
	border	[EtchedBorder]	
	borderPainted		
	componenti opapiliena	sinene's	v
	contentAreaFilled		
		e f he	
	House_1RB [JRadioButton]		(?)
		Clo	se Help

Εικόνα 8: Ιδιότητες border και borderPainted

Επαναλαμβάνουμε το βήμα 8 και για τα άλλα δύο radioButton σύμφωνα με την εκφώνηση. Ολοκληρώνουμε τη σχεδίαση της διεπαφής προσθέτοντας ένα jButton και άλλα δύο jLabel στον καμβά κάτω από το πρώτο jPanel όπως φαίνεται στην Εικόνα 9.



Εικόνα 9: Ολοκλήρωση της σχεδίασης της διεπαφής

Βήμα 9: Σε αυτό το βήμα θα προγραμματίσουμε τι θα συμβαίνει όταν ο χρήστης της εφαρμογής επιλέγει ένα από τα τρία σπίτια (radioButtons) με το αριστερό πλήκτρο του ποντικιού. Πολλά event μπορούν να προγραμματιστούν για το συγκεκριμένο γεγονός. Το πιο κατάλληλο για να διαχειριστούμε radiobuttons είναι το *item->itemStateChanged*.

Έτσι, επιλέγουμε να προγραμματίσουμε για το πρώτο radiobutton το συγκεκριμένο event (Εικόνα 10). Θα πρέπει να γίνουν δύο ενέργειες όταν εκτελείται το συγκεκριμένο γεγονός. Πρώτον να εμφανίζεται στην ετικέτα το ποσό του σπιτιού και δεύτερον σε μία global μεταβλητή price να αποθηκεύεται η τιμή του σπιτιού. Αυτή τη global μεταβλητή θα τη χρησιμοποιήσουμε αργότερα. Στην Εικόνα 11, φαίνεται ο κώδικας των τριών radiobutton για το συγκεκριμένο γεγονός ενώ στη γραμμή 281 δηλώνεται η global μεταβλητή. Τέλος, αλλάζουμε το text της ετικέτας AksiaTimiLabel σε 3.000.000 για να αντιστοιχεί στην τιμή του πρώτου σπιτιού (Εικόνα 12).

Start Page X Ergastirio3.java X			🕕 🔻 🗆 🏼 Pal	lette
Source Design History ID ID	Edit Text Change Variable Name]		Swin Pane Split Tool Inte Swin Labe Tog Rad
Υπολογιαμός Δόσης Δόση ανά μήνα: 0	Bind > Events > Align = Anchor = Auto Resizing = Same Size = Set to Default Size = Enclose In = 2 Edit Layout Space	Action > Ancestor > Change > Component > Container > Focus > Hierarchy8ounds > InputMethod >	Ho Pr P acti bac moc but for for	Com use oper ion kgro del tton(t egrou
AGo:	Design Parent	ltem 2	itemStateChang	jed
Duriput X	Move Up Move Down Cut Copy Duplicate	Key > Mouse > MouseMotion > MouseWheel > PropertyChange > VetoableChange >	v	

Εικόνα 10: Item – itemStateChanged γεγονός



Εικόνα 11: Ο κώδικας των τριών radiobutton για το γεγονός itemStateChanged

Design Histor	Y HV E	
Στοιχεία Δανείου		Γ Διαθέσιμα Σπίτια
Κεφάλαιο:	0	
Εππόκιο:	0	1
Διάρκεια:	0 μήνες	
Υπολογισμός Δά	ianc	
Thomos popos, as		-t-t-
ιόση ανά μήν <mark>α</mark> :	0	to a the second
		Alia: 3.000.000

Εικόνα 12: Το text του JLabel AksiaTimiLabel παίρνει την τιμή του πρώτου radiobutton

Βήμα 10: Όταν ο χρήστης θα μεταφέρει το δείκτη του ποντικιού μέσα σε ένα radiobutton χωρίς να πατήσει κάποιο πλήκτρο θα πρέπει να αλλάζει το text της ετικέτας AksiaTimiLabel στην τιμή του συγκεκριμένου σπιτιού που απεικονίζεται στο radiobutton. Αυτό γίνεται προγραμματίζοντας το event MouseEntered σε κάθε ένα από τα τρία radiobutton, όπως φαίνεται στην Εικόνα 13. Ενώ όταν θα μεταφέρει το δείκτη του ποντικιού έξω από το radiobutton θα πρέπει να αλλάζει το text της ετικέτας AksiaTimiLabel στην τιμή του επιλεγμένου (selected) σπιτιού της ομάδας. Αυτό γίνεται προγραμματίζοντας το event MouseExited σε κάθε ένα από τα τρία radiobutton, όπως φαίνεται στην Εικόνα 14. Για να υλοποιήσουμε αυτό το γεγονός χρησιμοποιήσουμε τη μεταβλητή price, στην οποία είναι αποθηκευμένη η τιμή του επιλεγμένου σπιτιού.

Start Page	e ×	🔂 Ergastirio3.java 🗙
Source	Design	History 📴 💀 - 🐺 - 🗖 - 🖓 😓 🖓 😓 🍄 😓 🏷 😓 💷 😏 😑 🔲 些 🚅
261 📮	2	<pre>private void House_1RBMouseEntered(java.awt.event.MouseEvent evt) {</pre>
262		<pre>AksiaTimiLabel.setText("3.000.000");</pre>
263	L	}
264		
265 📮	2	<pre>private void House_2RBMouseEntered(java.awt.event.MouseEvent evt) {</pre>
266		<pre>AksiaTimiLabel.setText("5.000.000");</pre>
267	L	}
268		
269 📮	2	<pre>private void House_3RBMouseEntered(java.awt.event.MouseEvent evt) {</pre>
270		<pre>AksiaTimiLabel.setText("4.000.000");</pre>
271	L	}
272		



Start P	age	× 🔂 Ergastirio3.java ×
Source	e De	esign History 🔯 👼 - 👼 - 💐 - 💐 🖓 - 🖓 - 🖓 - 🥵 - 🏷 - 🏷 - 🎽 - 🖉 - 📕 - 🖉 - 📕
201		
282	F	<pre>private void House_IRBMouseExited(java.awt.event.MouseEvent evt) {</pre>
283		AksiaTimiLabel.setText(price);
284	L	}
285		
286	F	<pre>private void House_2RBMouseExited(java.awt.event.MouseEvent evt) {</pre>
287		AksiaTimiLabel.setText(price);
288	L	}
289		
290	Ę	<pre>private void House_3RBMouseExited(java.awt.event.MouseEvent evt) {</pre>
291		AksiaTimiLabel.setText(price);
292	L	}
293		
294	Ģ	/**
295		* Anaram args the command line arguments

Εικόνα 14: Ο κώδικας των τριών radiobutton για το γεγονός MouseExited

Βήμα 11: Τώρα θα προγραμματίσουμε το γεγονός ActionPerformed του κουμπιού "Υπολογισμός Δόσης" για να βρούμε τη δόση δανείου που θα πληρώσει ο πελάτης σύμφωνα με τις τέσσερις παραμέτρους, δηλαδή την αξία σπιτιού, το κεφάλαιο που έχει ο πελάτης (τα χρήματα που έχουν), το επιτόκιο δανεισμού και τον αριθμό των δόσεων που επιθυμεί ο πελάτης. Στην Εικόνα 15, παρουσιάζεται ο κώδικας υπολογισμού της δόσης.

Στη γραμμή 303, η μεταβλητή *kefalaioD* χρησιμοποιείται για να «κρατάει» το κεφάλαιο που έχει δηλώσει ο πελάτης ότι έχει στο TextField *KefalaioText*. Στο TextField έχει εισαχθεί η τιμή ως κείμενο. Θεωρούμε ότι το *KefalaioText* δεν είναι κενό και περιέχει αριθμό. Σε περίπτωση που ο αριθμός έχει γραφτεί σε μορφή «Ν.ΝΝΝ.ΝΝΝ», δηλαδή με χρήση της τελείας για διαχωρισμό των ψηφίων τότε με την μέθοδο replace αφαιρούμε την τελεία. Τέλος πριν αποθηκευτεί το ποσό του κεφαλαίου που δηλώθηκε στο *KefalaioText* στη μεταβλητή *kefalaioD* μετατρέπεται σε double.

Οι επόμενες τρεις γραμμές του κώδικα είναι παρόμοιες για το επιτόκιο του δανείου, τη διάρκεια της δόσης σε μήνες και την αξία του σπιτιού.

Στη γραμμή 308 βρίσκουμε το δάνειο που χρειάζεται ο πελάτης. Αν το δάνειο είναι αρνητικό σημαίνει ότι ο πελάτης έχει τα λεφτά για να αγοράσει το σπίτι και έτσι εμφανίζουμε κατάλληλο μήνυμα (γραμμή 310) στην ετικέτα *EtiketaEnimerosis* που προσθέτουμε στο JFrame σε κατάλληλο σημείο (Εικόνα 16). Αλλιώς, υπολογίζουμε τον τόκο (γραμμή 313), βρίσκουμε το συνολικό ποσό (δάνειο + τόκος) (γραμμή 314), και αν η διάρκεια έχει οριστεί μεγαλύτερη από το μηδέν, βρίσκουμε (γραμμή 318) και εμφανίζουμε τη δόση (γραμμή 319). Αν η διάρκεια δεν έχει οριστεί μεγαλύτερη του μηδενός εμφανίζουμε κατάλληλο μήνυμα (γραμμή 316).

Start P	Start Page × 🗈 Ergastirio3.java ×					
Source	2 1	Design	History 🛛 🚱 🗸 🐺 🖓 🖓 🤻 🎖 🖓 🖓 😓 🖓 🗐 🗐 🕘 🔲 🏙 🚅			
300	-		3			
301						
302	F	1	private void YpologismosButtonActionPerformed(java.awt.event.ActionEvent evt) {			
303			<pre>double kefalaioD = Double.valueOf(KefalaioText.getText().replace(".", ""));</pre>			
304			<pre>double epitokioD = Double.valueOf(EpitokioText.getText().replace(".", ""));</pre>			
305			<pre>double diarkeiaD = Double.valueOf(DiarkeiaText.getText().replace(".", ""));</pre>			
306			<pre>double axiaD = Double.valueOf(price.replace(".", ""));</pre>			
307						
308			double daneio = axiaD - kefalaioD;			
309			if (daneio <= 0)			
310			EtiketaEnimerosis.setText("Δεν χρειάζεσαι δάνειο!");			
311			else {			
312			<pre>EtiketaEnimerosis.setText("");</pre>			
313			<pre>double tokos = (daneio * epitokioD)/100;</pre>			
314			<pre>double daneiometoko = daneio + tokos;</pre>			
315			if (diarkeiaD<=0)			
316			DosiTimiLabel.setText("λάθος διάρκεια");			
317			else {			
318			<pre>double dosi = daneiometoko/diarkeiaD;</pre>			
319			<pre>DosiTimiLabel.setText(String.valueOf(dosi));</pre>			
320			}			
321			}			
322	L		}			
323						

Εικόνα 15: Ο κώδικας υπολογισμού της δόσης

🎒 Αγορά Σπιτιού			<u> </u>	
Στοιχεία Δανεί	ίου		Διαθέσιμα Σπίτια	
Κεφάλαιο:	10.000.000		· · · · · · · · ·	
Επιτόκιο:	10			
Διάρκεια:	100	μήνες		
Υπολογισμός Δόση ανά μήνα: Δεν χρειάζεσαι δ	<u>Δόσης</u> 0 άνειο!		Afa: 3000	

Εικόνα 16: Εμφάνιση κατάλληλου μηνύματος αν ο πελάτης δεν χρειάζεται δάνειο.

Βήμα 12: Στο τελευταίο βήμα της άσκησης θα ορίσουμε τις συντομεύσεις. Με αυτές θα μπορεί ο χρήστης να εκτελέσει κάποιες ενέργειες με συνδυασμό πλήκτρων από το πληκτρολόγιο.

Έτσι, για να ορίσουμε συντόμευση σε ένα TextField θα πρέπει να οριστεί η συντόμευση σε μία ετικέτα (JLabel). Όπως φαίνεται στην Εικόνα 17, για να ορίσουμε τη συντόμευση ALT+K για το

KefalaioText θα πρέπει να ορίσουμε τρεις ιδιότητες της ετικέτας *KefalaioLabel* που προσδιορίζει το *KefalaioText*. Την ιδιότητα *displayedMnemonic*, όπου ορίζουμε το γράμμα (K) που θα συνδυάσουμε με το πλήκτρο ALT (default) – προσοχή αν ορίσουμε ελληνικό γράμμα θα πρέπει το πληκτρολόγιο να είναι ελληνικό όταν θα χρησιμοποιούμε την συντόμευση. Την ιδιότητα *labeFor*, όπου ορίζουμε τη μεταβλητή του TextField που θέλουμε να μεταφερθεί ο cursor με το συνδυασμό των πλήκτρων ALT+K. Τέλος, την ιδιότητα *displayedMnemonicIndex*, όπου ορίζουμε ποιο γράμμα θα υπογραμμιστεί στο JLabel για να δείξουμε στο χρήστη πως θα χρησιμοποιήσει τη συντόμευση – ορίζουμε έναν αριθμό, ξεκινώντας από το μηδέν προσδιορίζοντας το γράμμα του κειμένου της ετικέτας που θέλουμε να υπογραμμιστεί. Δηλαδή, για να υπογραμμιστεί το πρώτο γράμμα χρησιμοποιούμε το μηδέν, για το δεύτερο το ένα κτλ. Με παρόμοιο τρόπο ορίζουμε τις συντομεύσεις και στα άλλα δύο TextFields (Εικόνα 18 και Εικόνα 19).

Για το JButton *YpologismosButton* αρκεί να ορίσουμε τις ιδιότητες *mnemonic* και *displayedMnemonicIndex* όπως φαίνεται στην Εικόνα 20.

Τέλος, για τα radioButton που περιέχουν μόνο εικόνα δεν χρειάζεται να οριστεί η ιδιότητα displayedMnemonicIndex. Αρκεί να οριστεί κατάλληλα η ιδιότητα mnemonic όπως φαίνεται στην Εικόνα 21, για το πρώτο radioButton.

Debug Profile Team Tools Window	H 🔘 KefalaioLabel [JLabel] - Properti	es	×
50 50 N	Properties Binding Events	Code	
nmg> 🗸 👔 🚱 🕨 🖬 🗊	Properties		^
Start Page X Regasting 3 java	background	[240,240,240]	
	displayedMnemonic	к	
Source Design History	font	Tahoma 11 Plain	
	foreground	[0,0,0]	
	horizontalAlignment	LEADING	~
Στοιχεία Δανείου	icon	<none></none>	~
ζεφάλαιο: 0	labelFor	KefalaioText	~
Entróxio: 0	text	Κεφάλαιο:	
	toolTipText		
Διάρκεια: 0	ur verticalAlignment	CENTER	~
	Other Properties		
	UIClassID	LabelUI	
	alignmentX	0.0	
	alignmentY	0.5	
Υπολογισμός Δόσης	autoscrolls		
	baselineResizeBehavior	CENTER_OFFSET	\sim
Δόση ανά μήνα: Ο	border	(No Border)	
	componentPopupMenu	<none></none>	~
	cursor	Default Cursor	~
	debugGraphicsOptions	NO_CHANGES	~
	disabledicon	<none></none>	~
	displayedMnemonicIndex	0	🗸
	KefalaioLabel [JLabel]	Close	Help

Εικόνα 17: Ορισμός συντομεύσεων για το KefalaioText

urce	Design	History	-	rioperdes	ounding	Events	 Jue		
				Properties					
				background			[240,240,240]		
	Στοιχεία Δ	ανείου		displayedMn	emonic		E		
	Κεφάλαιο:			font			Tahoma 11 Plain		4
			_	foreground			[0,0,0]		Ŀ
	=пітокіо:	-	-	horizontalAlign	ment		LEADING	~	
	Néovene		-	icon			<none></none>	~	
	μισρκεία.		-	labelFor			EpitokioText	~	
				text			Επτόκιο:		
				tooITipText					
				verticalAlignme	nt		CENTER	~	
	Υπολογια	σμός Δόσης		Other Proper	ties				
L				UIClassID			LabelUI		Į
Δόσο ανά μόνα: Ο		alignmentX			0.0				
				alignmentY			0.5		
				autoscrolls					[
				baselineResize	Behavior		CENTER_OFFSET		
				border			(No Border)		Į.
				componentPop	upMenu		<none></none>	~	
				cursor	ursor	Default Cursor	~		
		debugGraphics	Options		NO_CHANGES	~			
		disabledicon			<none></none>	~			
		displayedMn	emoniclno	lex	0				
			EpitokioLabe	[JLabel]		_			

Εικόνα 18: Ορισμός συντομεύσεων για το EpitokioText

ource Design History	Properties Binding Events	Code	
	Properties		
	background	[240,240,240]	
Στοιχεία Δανείου	displayedMnemonic	D	
Κεφάλαιο:	font	Tahoma 11 Plain	
	foreground	[0,0,0]	
Επιτόκιο:	horizontalAlignment	LEADING	v
	icon	<none></none>	v
Διορκεια:	labelFor	DiarkeiaText	~
5	text	Διάρκεια:	
	toolTipText		
	verticalAlignment	CENTER	v
Υπολογισμός Δόσης	Other Properties		
there here a sould	UIClassID	LabelUI	
Λόσο ανά μόνα: Ο	alignmentX	0.0	
Tool and billion o	alignmentY	0.5	
	autoscrolis		
	baselineResizeBehavior	CENTER_OFFSET	
	border	(No Border)	
	componentPopupMenu	<none></none>	v
	cursor	Default Cursor	~
	debugGraphicsOptions	NO_CHANGES	~
	disabledicon	<none></none>	~
	displayedMnemonicIndex	0	
	DiarkeiaLabel [JLabel]		se Help

Εικόνα 19: Ορισμός συντομεύσεων για το DiarkeiaText

ource Design His	tory 🛛 🗔 🐺	action	<none></none>	~
		background	[240,240,240]	
		font	Tahoma 11 Plain	
Στοιχεία Δανεί	00	foreground	[0,0,0]	
Κεφάλαιο:	0	icon	<none></none>	× .
		mnemonic	Y	
Επιτόκιο:	0	text	Υπολογισμός Δόσης	
6.5mm	0	toofTipText		
Διαρκεία:	0	Other Properties		
		UIClassID	ButtonUI	
		actionCommand	Υπολογισμός Δόσης	
		alignmentX	0.0	
Υπολογισμός	Δόσης 👂	alignmentY	0.5	
0 0		autoscrolis		[.
Δόση ανά μήνα:	0	baselineResizeBehavior	CENTER_OFFSET	
		border	[XPEmptyBorder]	
		borderPainted		
		buttonGroup	<none></none>	~ .
		componentPopupMenu	<none></none>	× .
		contentAreaFilled		
		cursor	Default Cursor	× .
		debugGraphicsOptions	NO_CHANGES	× .
		defaultCapable		
		disabledicon	<none></none>	× .
		disabledSelectedIcon	<none></none>	~ .
		displayedMnemonicIndex	0	
		doubleBuffered		
		YpologismosButton [JButton]		

Εικόνα 20: Ορισμός συντομεύσεων για το YpologismosButton

Start Pa	ge × 🗟	Ergastirio3.	House_1RB [JRadioButton]	- Properties	×	;
Source	Design	History	Properties Binding Events	s Code		
			Properties			~
			action	<none></none>	~	
	- Στοιχεία Δα	veiou	background	[240,240,240]		
	Κεφάλαιο:		model	<default></default>	v	
			buttonGroup	buttonGroup1	v	
	Επιτόκιο:		font	Tahoma 11 Plain		
		_	foreground	[0,0,0]		
	Διαρκεια:		mnemonic	1		
			selected	\checkmark		
			text			
			toolTipText			
	Υπολογισ	υός Δόσης	Other Properties			
			UIClassID	RadioButtonUI		
	Δόση ανά μή	va: 0	actionCommand			
			alignmentX	0.0		
			alignmentY	0.5		
			autoscrolis			
			baselineResizeBehavior	CENTER_OFFSET		
			border	[EtchedBorder]		
			borderPainted	\sim		
			componentPopupMenu	<none></none>	×	
			contentAreaFilled	\checkmark		
				0.000		
			mnemonic		6)
			(int) the keyboard character mnem	onic		
					Close Help	

Εικόνα 21: Ορισμός συντομεύσεων για το House_1RB