

CEID

Οντοκεντρικός Προγραμματισμός I (Java)

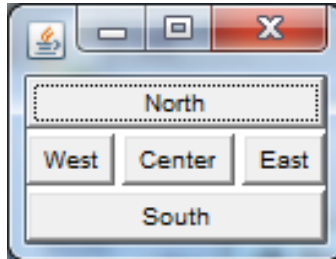
Ακαδ. Έτος: 2012-2013

Αναστασία Παπαρροδοπούλου AM : 3873

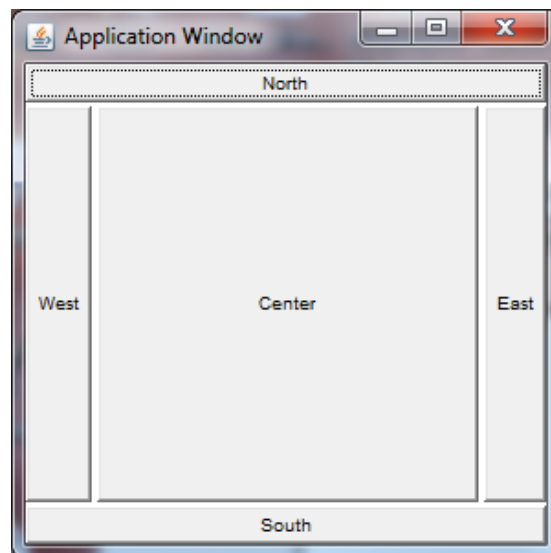
4/26/2013

7η Εργαστηριακή Άσκηση

1) α. Μετά τη μετάφραση και την εκτέλεση του προγράμματος πήραμε το εξής αποτέλεσμα:



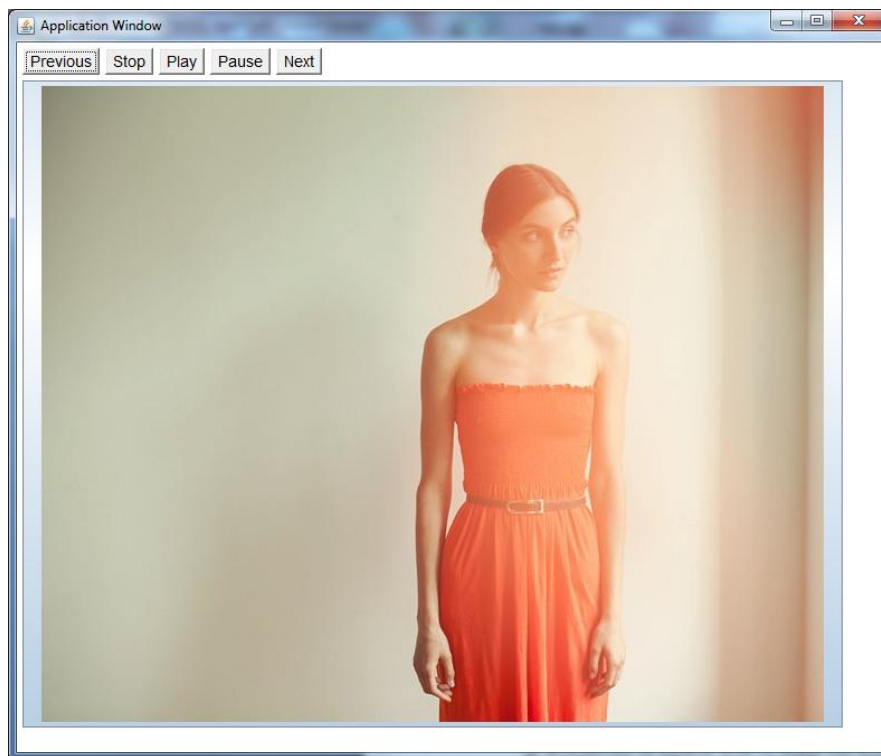
β. Βγάζοντας τα σχόλια από τη main και επανεκτελώντας το, παίρνουμε το εξής:



2) α. Τρέχοντας το νέο πρόγραμμα παίρνουμε τα εξής αποτελέσματα:



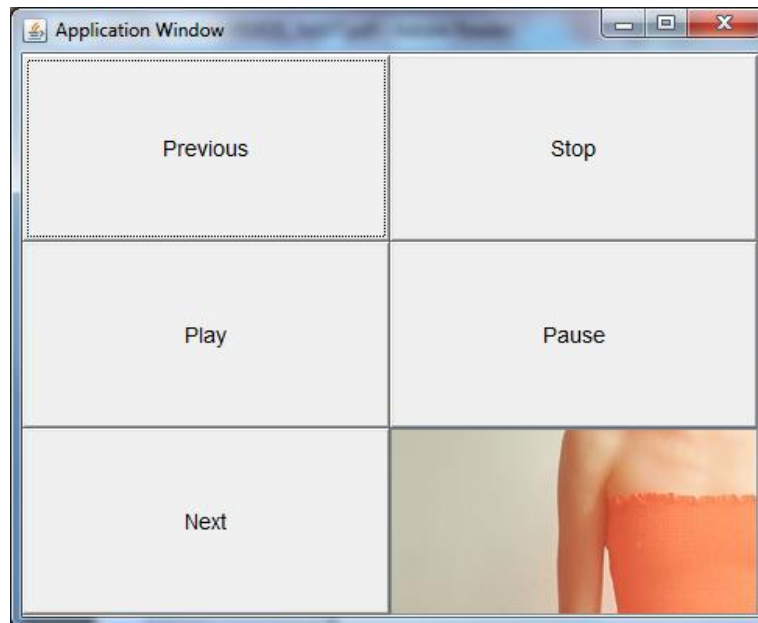
Προσπαθώντας να κάνουμε resize το παράθυρο, διαπιστώνουμε το εξής:



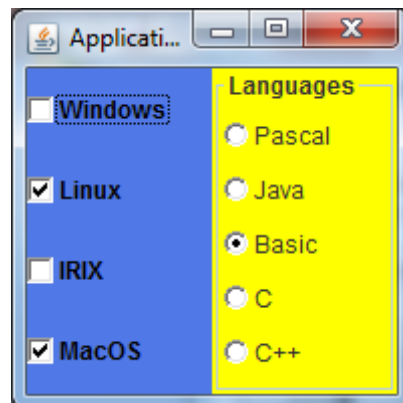
β. Βγάζοντας τα σχόλια από τη main και επανεκτελώντας το πρόγραμμα έχουμε:



Προσπαθώντας τώρα να κάνουμε `resize` του παραθύρου που δημιουργήθηκε έχουμε:



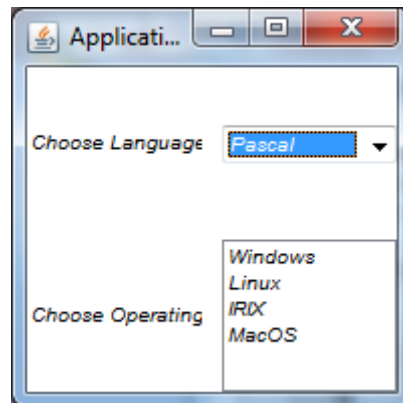
3) Τρέχοντας τη νέα κλάση παίρνουμε τα παρακάτω αποτελέσματα:



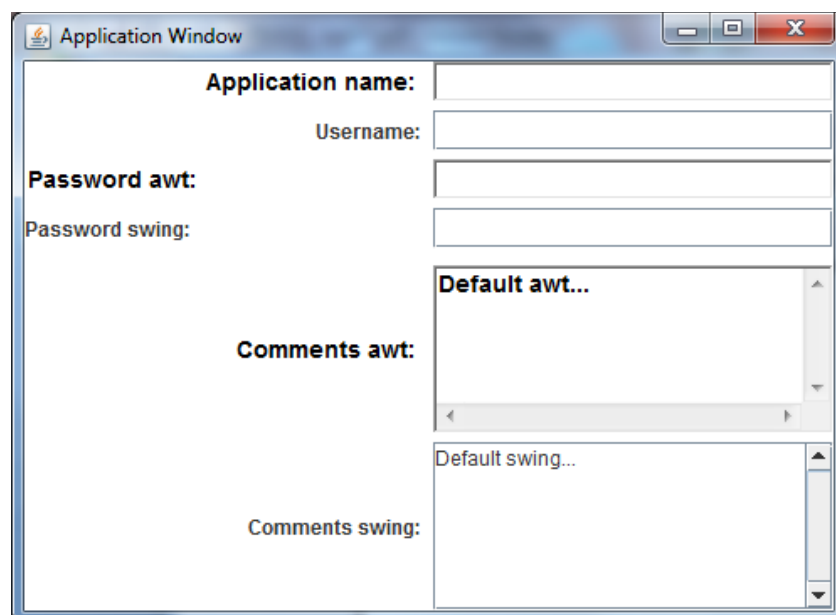
Παρατηρούμε ότι το πρώτο panel είναι τύπου `Panel`, έχει χρώμα μπλε και τα κουτιά επιλογής είναι έτσι ενσωματωμένα ώστε να επιτρέπεται η επιλογή περισσότερων από ένα. Εν αντιθέσει με το δεύτερο, το οποίο είναι τύπου `JPanel`, έχει χρώμα κίτρινο και επιτρέπει την επιλογή μόνο ενός από τα radio button.

Όσον αφορά το στιγμιότυπο, παρατηρούμε ότι δημιουργείται ένα για το κάθε είδος panel και αυτά εμφανίζονται μαζί χάρη στην ενσωμάτωσή τους στο ίδιο `Frame`.

4) Έπειτα από την εκτέλεση της νέας κλάσης παίρνουμε το παρακάτω αποτέλεσμα:

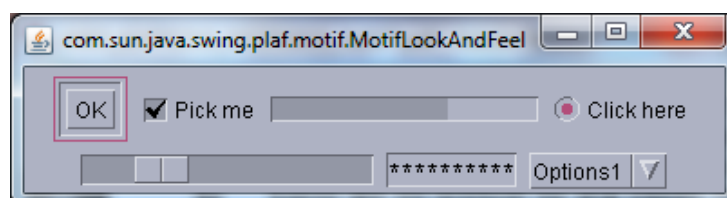


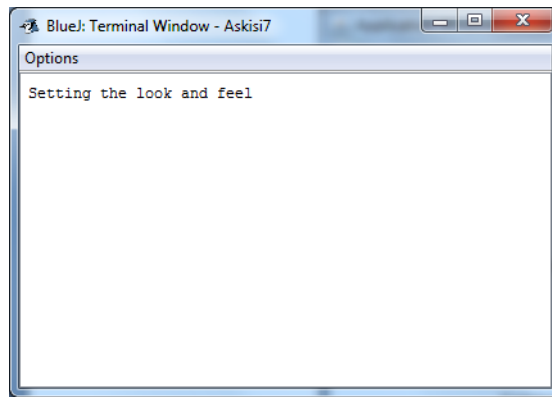
5) Έπειτα από την εκτέλεση της νέας κλάσης παίρνουμε το εξής αποτέλεσμα:



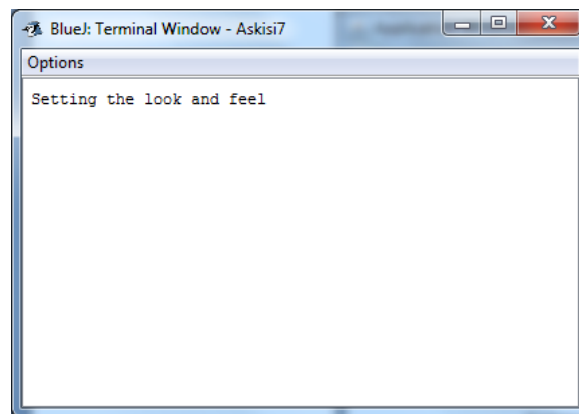
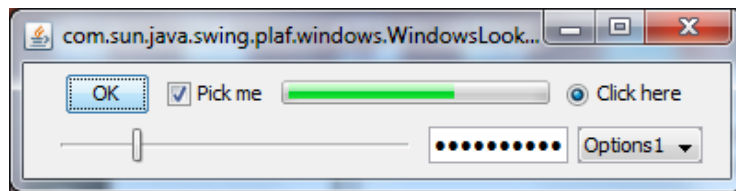
6) Μεταφράζοντας και εκτελώντας την κλάση αυτού το ερωτήματος, 4 φορές συνολικά, με τα ορίσματα “1”, “2”, “3”, καθώς και χωρίς όρισμα, παίρνουμε με την αντίστοιχη σειρά τα αποτελέσματα:

Με όρισμα “1”

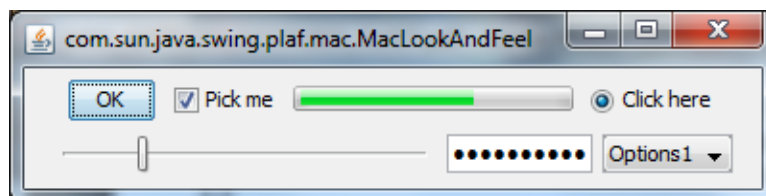


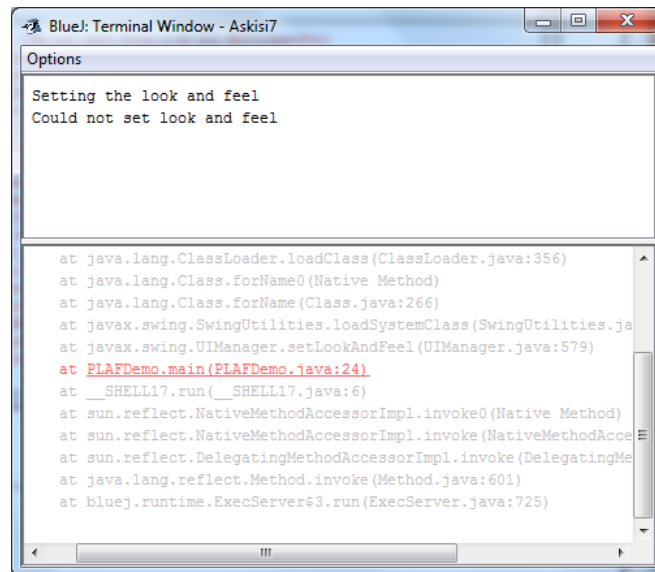


Με όρισμα “2”

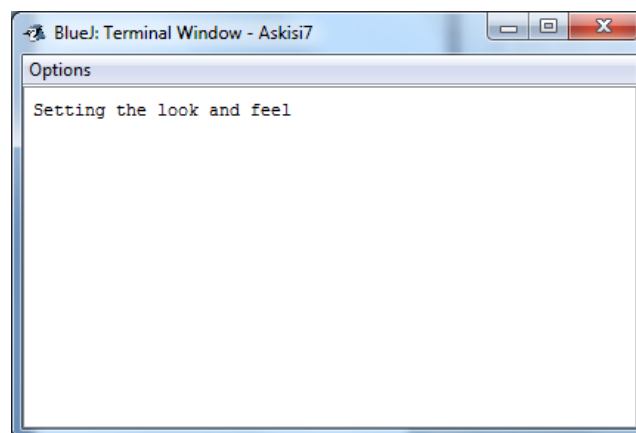
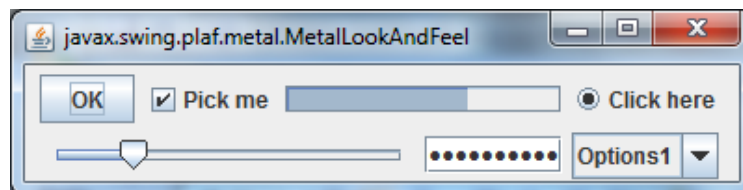


Με όρισμα “3” προέκυψε το παρακάτω σφάλμα και η εκτύπωση του ανάλογου μηνύματος, χωρίς να δημιουργηθεί κάποιο παράθυρο. Αυτό οφείλεται στο γεγονός ότι το MacLookAndFeel μπορεί να χρησιμοποιηθεί μονάχα σε πλατφόρμες και λειτουργικά του Mac OS, και όχι σε Windows, όπου και εκτελείται η παρούσα άσκηση.

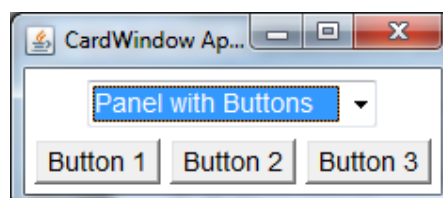




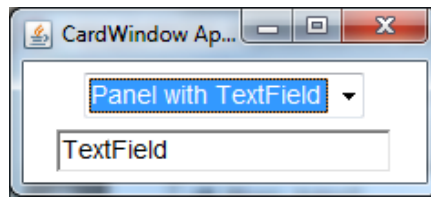
Με κανένα όρισμα



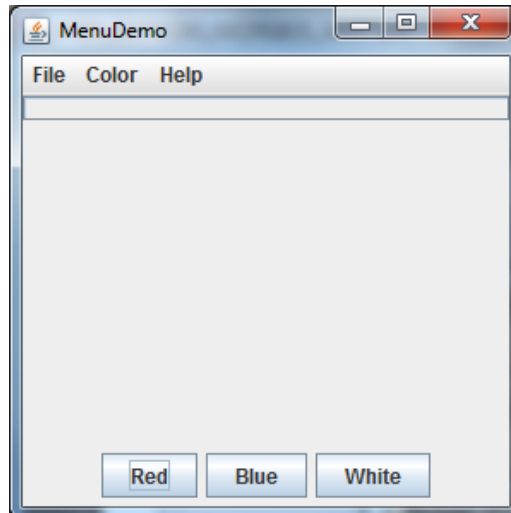
7) α. Μεταφράζοντας και εκτελώντας την κλάση αυτή έχουμε τα ακόλουθα αποτελέσματα:



β. Αλλάζοντας την τιμή στη λίστα επιλογής, παρατηρούμε πως αλλάζει και το περιεχόμενο του παραθύρου, όπως φαίνεται και στην εικόνα:



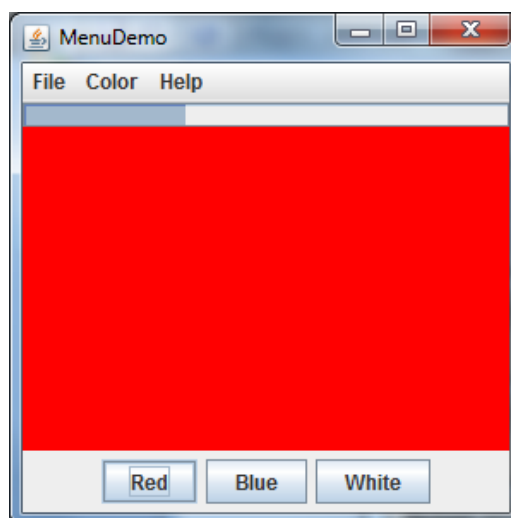
8) α. Τρέχοντας τη MenuDemo έχουμε τα εξής αποτελέσματα:



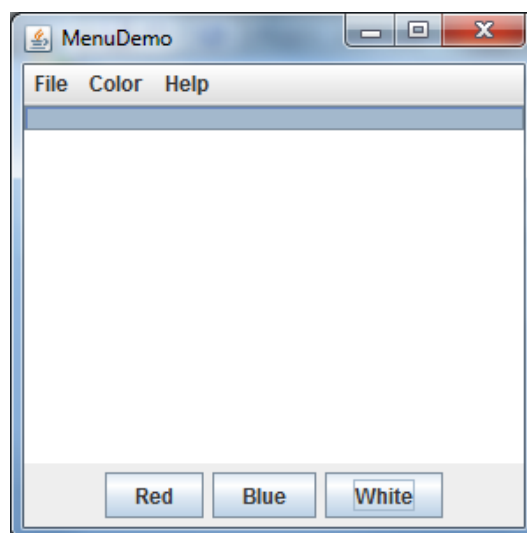
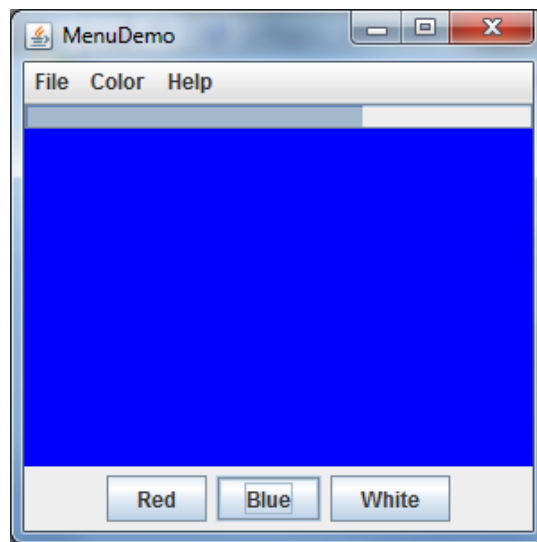
β. Αλληλεπιδρώντας με το menu και τα button παρατηρούμε τη λειτουργία της εφαρμογής:

Επιλέγοντας το "File->Exit" παρατηρούμε ότι κλείνει η εφαρμογή.

Πατώντας το κουμπί "Red" βλέπουμε ότι αλλάζει το χρώμα στη μεσαία περιοχή του παραθύρου, ενώ η μπάρα αμέσως κάτω από το κεντρικό menu γίνεται κατά το ένα τρίτο της γαλάζια.

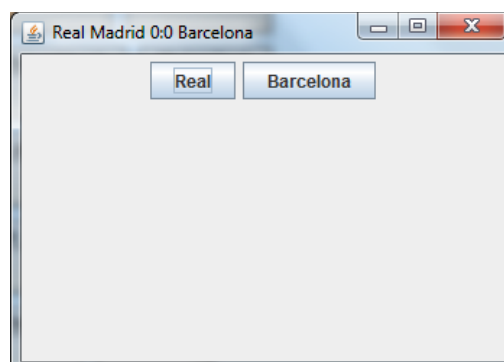


Αντίστοιχα, πατώντας το κουμπι “Blue”, γίνεται μπλε η εν λόγω περιοχή, ενώ πατώντας το κουμπι “White” γίνεται λευκή. Η μπάρα κάτω από το menu γίνεται κατά τα 2/3 γαλάζια και ολόκληρη γαλάζια, σε κάθε περίπτωση από τις παραπάνω.

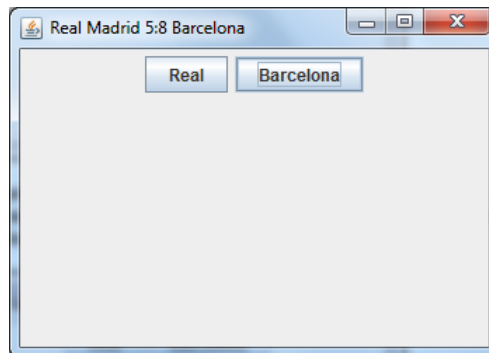


Τέλος, παρατηρούμε πως αν περιηγηθούμε λίγο στο menu, βλέπουμε ότι υπάρχουν και εκεί οι επιλογές για αλλαγή χρώματος της μεσαίας περιοχής, καθώς και ένα menu για βοήθεια σχετικά με την εφαρμογή.

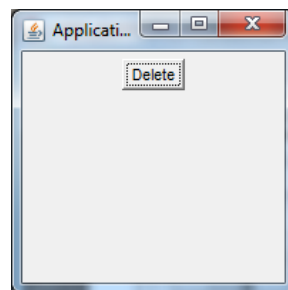
9) α. Τρέχοντας την δοθείσα εφαρμογή παίρνουμε τα εξής αποτελέσματα:



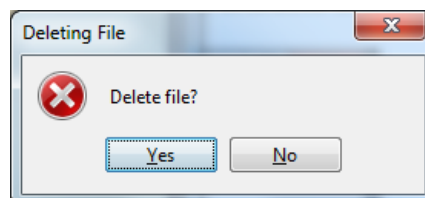
β. Πατώντας τα κουμπιά “Real” και “Barcelona” παρατηρούμε ότι αλλάζει το «σκορ» στον τίτλο του παραθύρου, κατά ένα για κάθε φορά που πατάμε το αντίστοιχο κουμπί. Έτσι, πατώντας τυχαία τα κουμπιά πήραμε το εξής αποτέλεσμα:



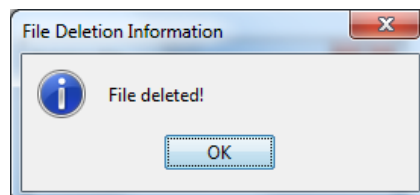
10) α. Τρέχοντας την εφαρμογή παίρνουμε τα ακόλουθα αποτελέσματα:



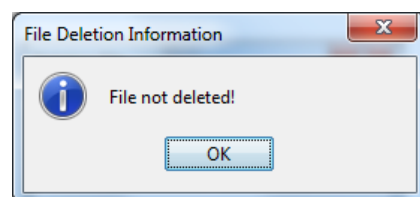
β. Πατώντας το κουμπί “Delete” εμφανίζεται το παρακάτω παράθυρο:



γ. Πατώντας “Yes” έχουμε το εξής αποτέλεσμα:



δ. Τέλος, ξανατρέχοντας την εφαρμογή και πατώντας “No” αυτή τη φορά, έχουμε:

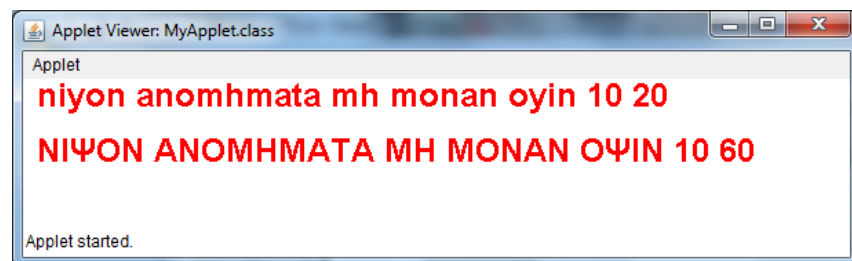


8η Εργαστηριακή Άσκηση

1) β. Μετά τη μετάφραση και την εκτέλεση της εφαρμογής πήραμε τα εξής αποτελέσματα:

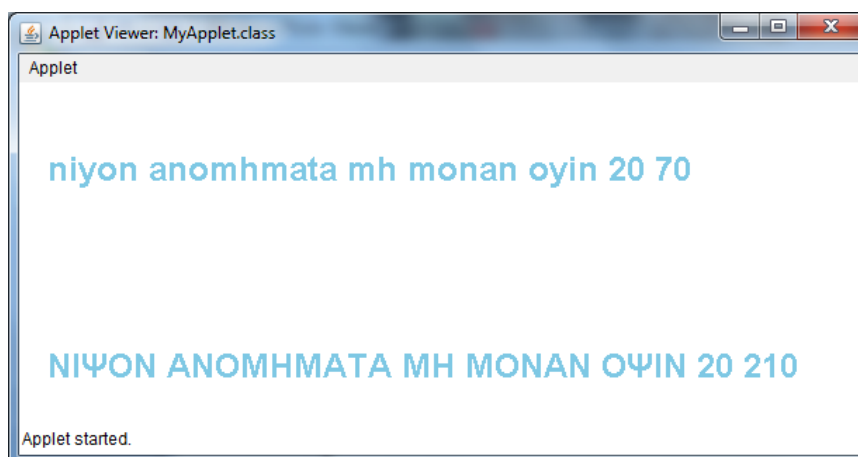


2) Τρέχοντας την εφαρμογή μετά την αλλαγή του κώδικά της, έχουμε:



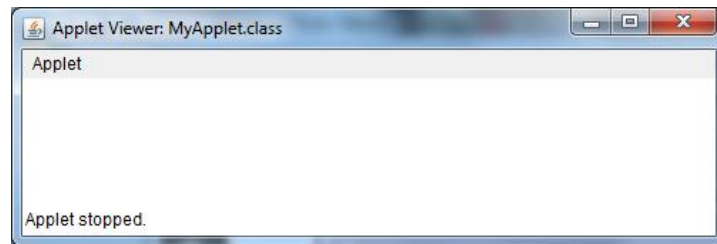
Παρατηρούμε ότι τώρα το κείμενο εκτυπώνεται με διαφορετική μορφοποίηση και όχι με αυτήν που είχε προηγουμένως.

3) α. Τρέχοντας την μικροεφαρμογή μετά από την προσθήκη του κώδικα, έχουμε:

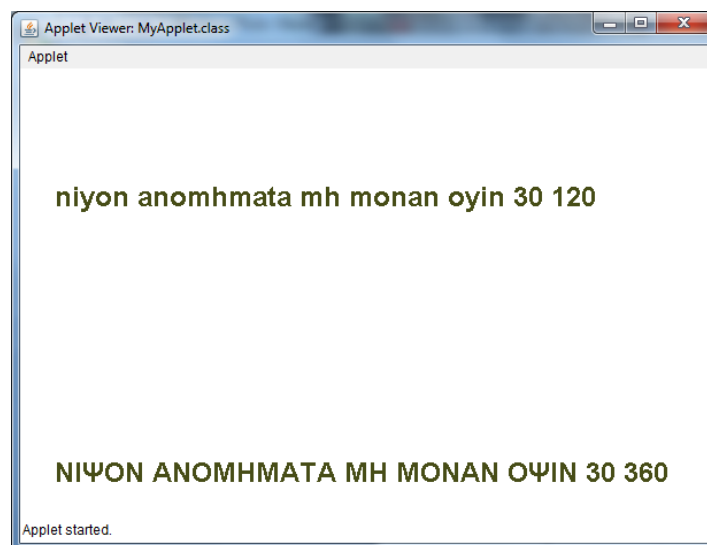


Παρατηρούμε ότι τα αποτελέσματα άλλαξαν, και συγκεκριμένα η μορφοποίηση του κειμένου.

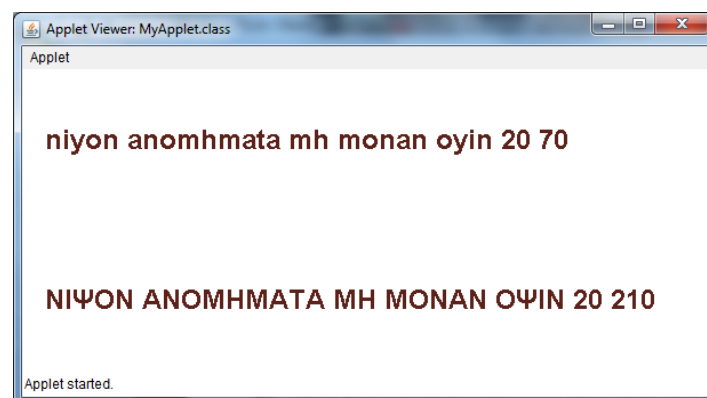
β. Επιλέγοντας το “Applet-Stop”, παρατηρούμε ότι το περιεχόμενο του παραθύρου παύει να υπάρχει, και η εφαρμογή μας σταματάει να εκτελείται.



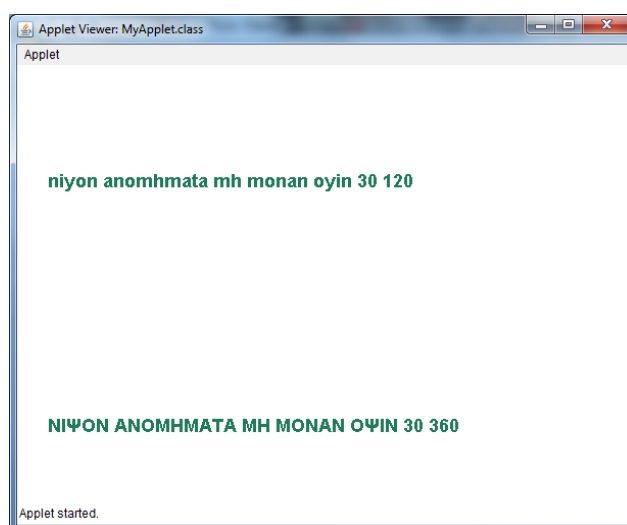
γ. Ξεκινώντας πάλι την εφαρμογή με το “Applet-Start” βλέπουμε ότι άλλαξε το χρώμα της γραμματοσειράς, καθώς εξαρτάται από μια τυχαία τιμή πλέον, η οποία προκύπτει κάθε φορά που εκτελούμε την εφαρμογή.



4) α. Μετά την προσθήκη του συγκεκριμένου κομματιού, παρατηρούμε ότι στην πρώτη εκτέλεση δεν αλλάζει κάτι πέρα από το χρώμα της γραμματοσειράς. Σύμφωνα με τον κώδικα που προσθέσαμε, θα πρέπει να περιμένουμε αλλαγή στην επόμενη εκτέλεση, και στο ερώτημα β.

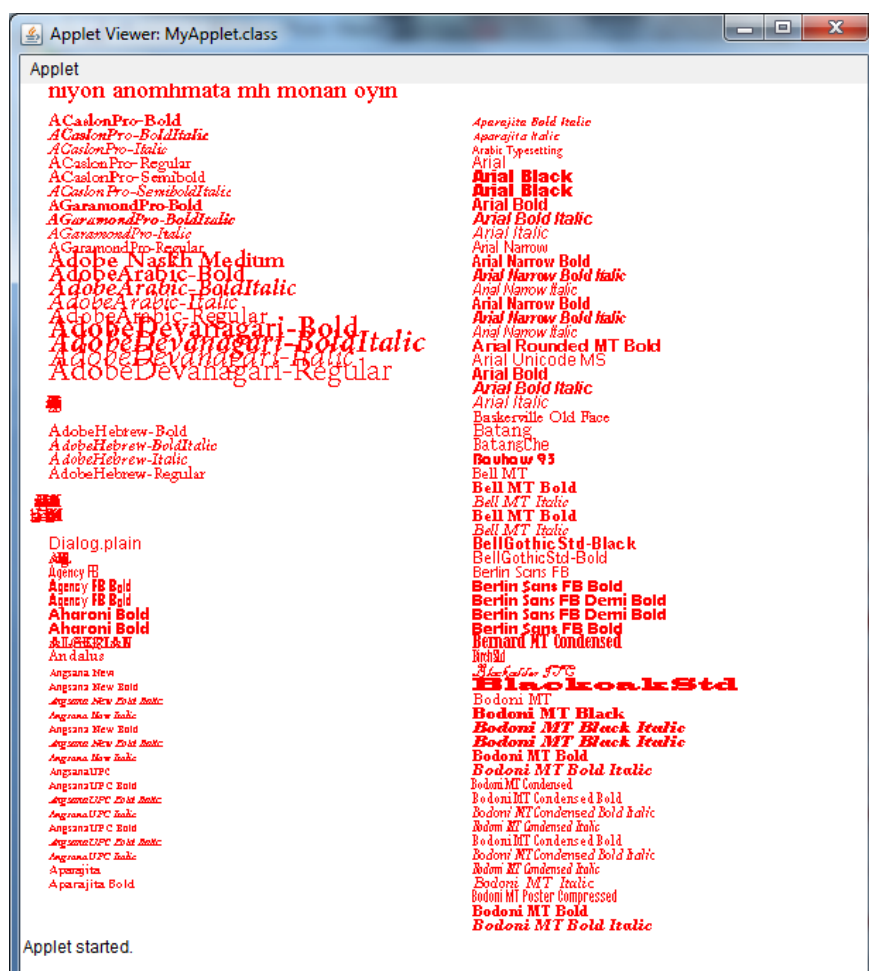


β. Σταματώντας την εφαρμογή και επαναξικινώντας την, έχουμε το εξής αποτέλεσμα:



Παρατηρούμε ότι τώρα το μήνυμα, εκτός από το διαφορετικό του χρώμα, εκτυπώνεται σε διαφορετική θέση απ' ότι το προηγούμενο και με μικρότερη γραμματοσειρά κατά 6 βαθμίδες, και αυτό οφείλεται στην συνθήκη που προσθήσαμε.

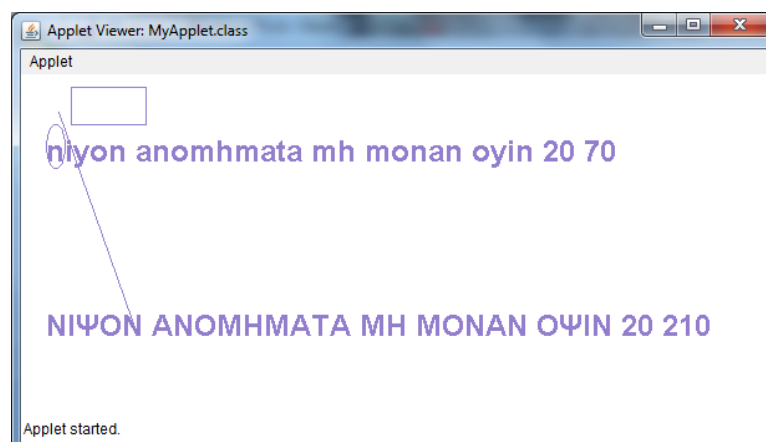
5) α. Εκτελώντας την μικροεφαρμογή και προσθέτοντας την παράμετρο που μας υπαγορεύει η εκκώνηση, με όνομα γραμματοσειράς Serif, παίρνουμε το εξής αποτέλεσμα:



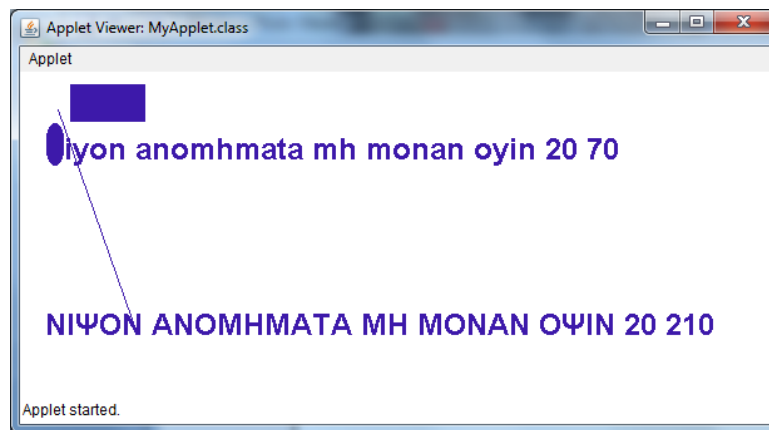
β. Αλλάζοντας τώρα το όνομα της παραμέτρου από font_name σε fontname, ουσιαστικά εκτυπώνονται πάλι οι δύο στήλες με τις 55 πρώτες γραμματοσειρές του υπολογιστή, το πάνω-πάνω κείμενο όμως εκτυπώνεται με την προσδιορισθείσα από τον κώδικα της εφαρμογής γραμματοσειρά, κι όχι με την Serif που προσδιορίσαμε εμείς.



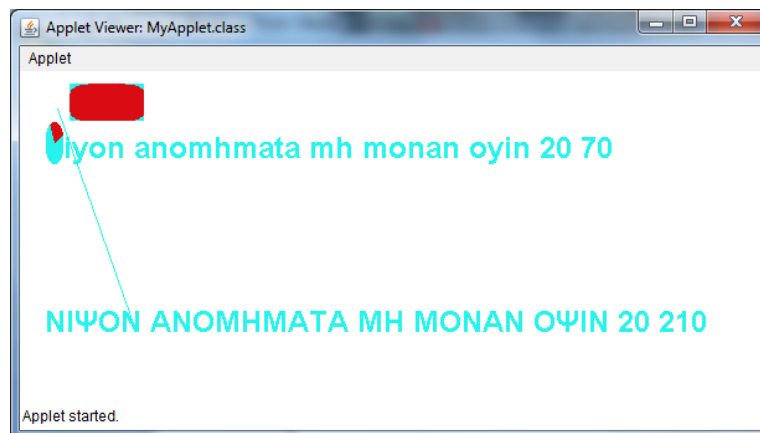
6) α. Τρέχοντας την εφαρμογή μετά από την αλλαγή, παίρνουμε τα εξής αποτελέσματα:



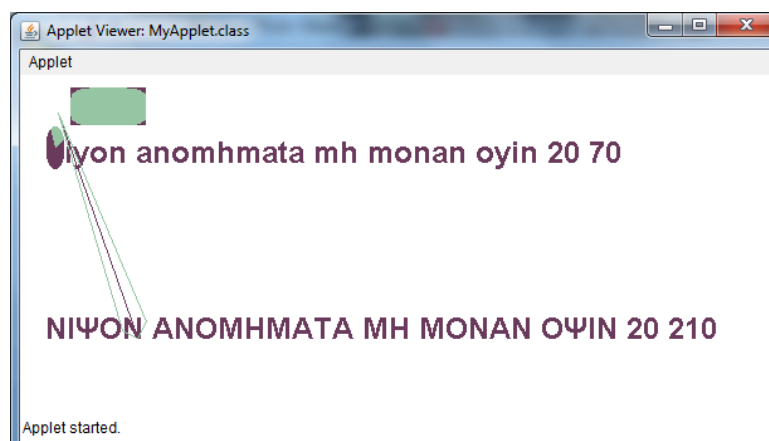
β. Αντικαθιστώντας τις δύο τελευταίες εντολές της paint με τις δοθείσες, έχουμε:



γ. Προσθέτοντας τον κώδικα που μας δίνεται, ουσιαστικά ζωγραφίζεται πάνω από το προηγούμενο οβάλ ένα κομμάτι μικρότερο από τεταρτημόριο και από το ορθογώνιο ένα οβάλ κομμάτι του. Είναι όπως φαίνεται παρακάτω:

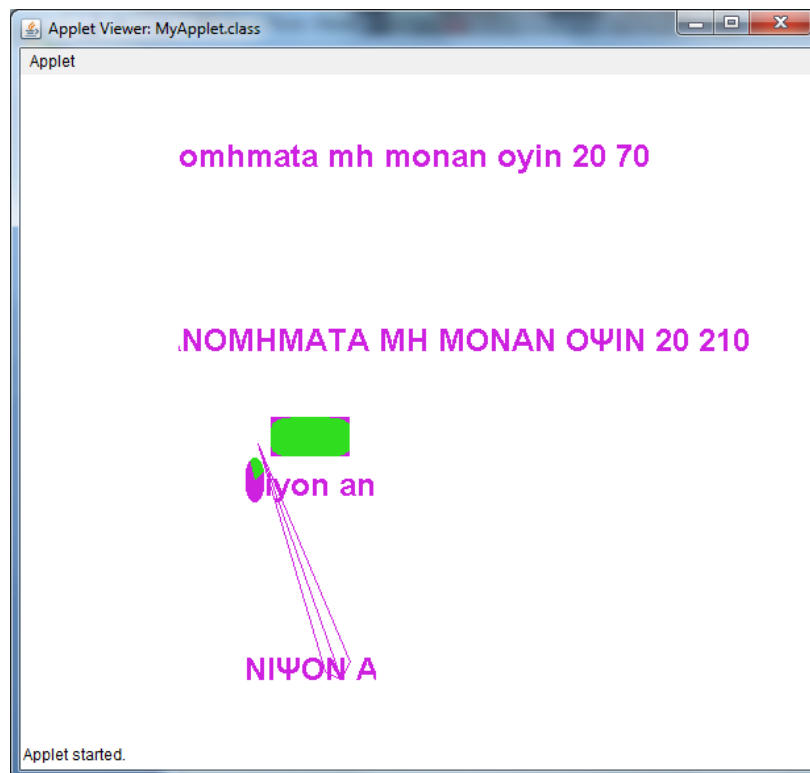


δ. Προσθέτοντας πάλι τον κώδικα που μας δίνεται, έχουμε τα εξής αποτελέσματα:

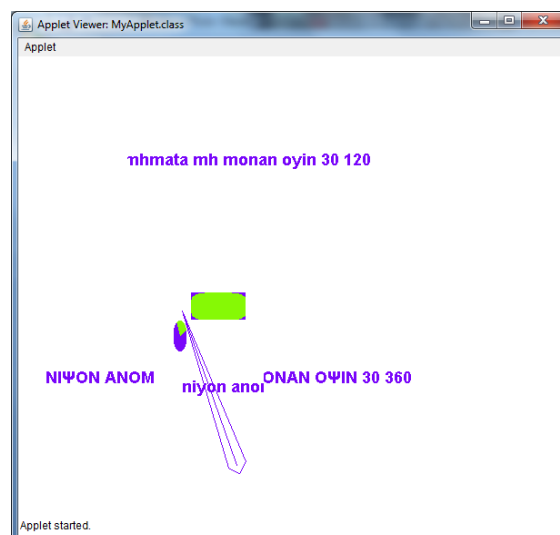


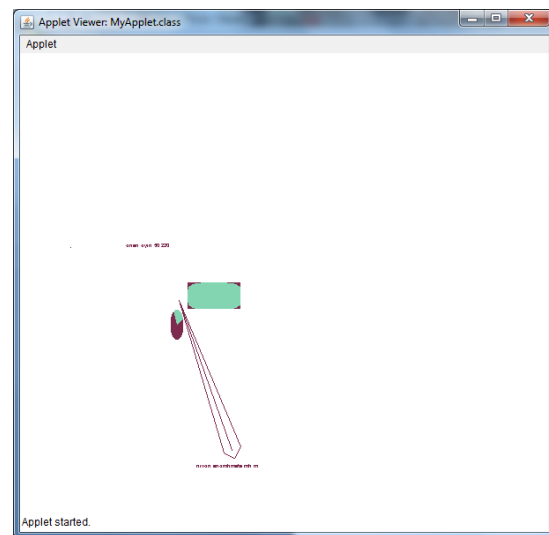
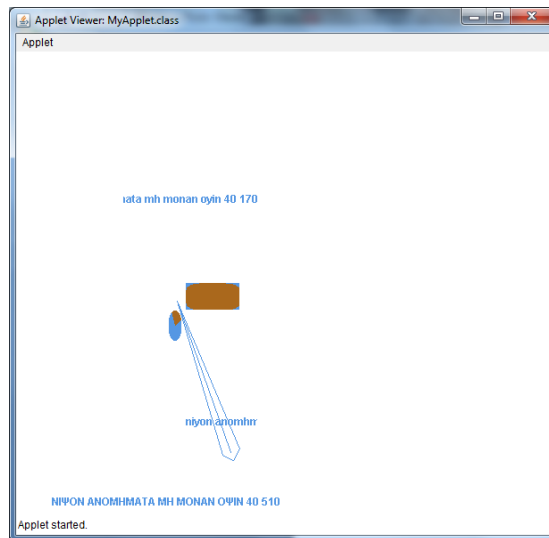
Ουσιαστικά εκτυπώνεται στην οθόνη ότι και πριν, με την προσθήκη ενός πολυγώνου με τέσσερις πλευρές και μια γραμμή που ενώνει δύο κορυφές του.

7) α. Αντικαθιστώντας την τελευταία εντολή της paint με τις δοθείσες, έχουμε:

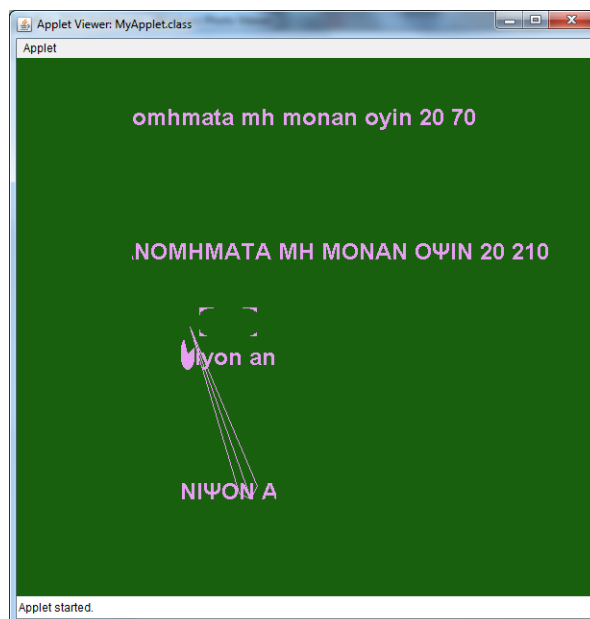


β. Σταματώντας και τρέχοντας εκ νέου την εφαρμογή παίρνουμε τα παρακάτω αποτελέσματα:

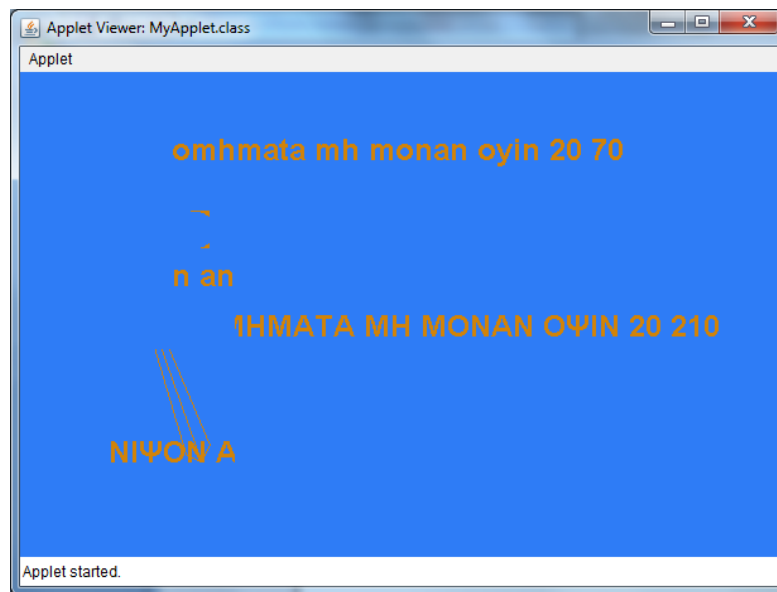




γ. Προσθέτοντας την εντολή που μας δίνεται, βλέπουμε ότι προστίθεται χρώμα (που προσδιορίστηκε παραπάνω στον κώδικα) στο παρασκήνιο της περιοχής όπου εκτυπώνεται το κείμενο:

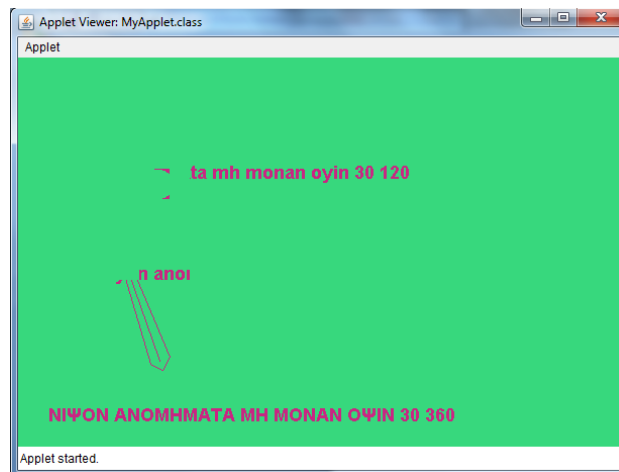


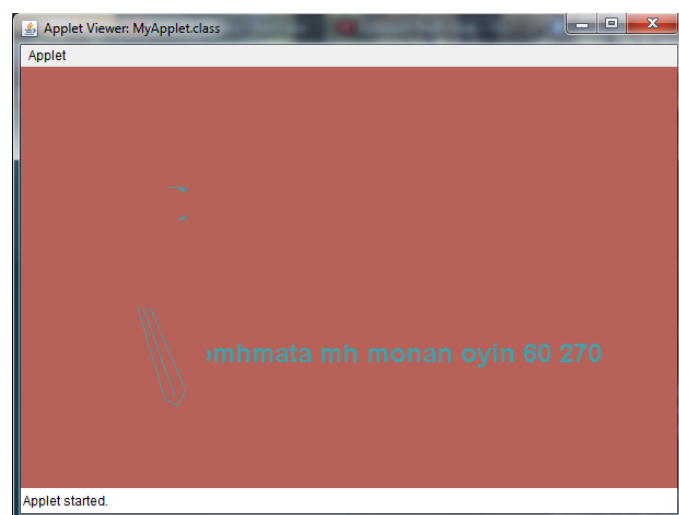
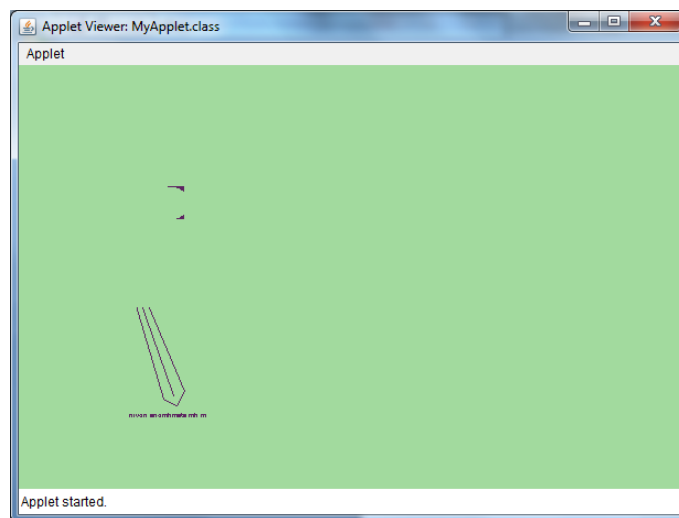
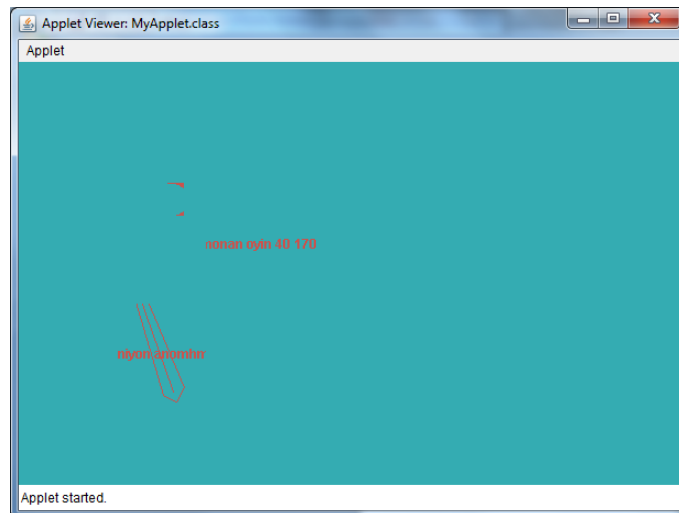
δ. Αλλάζοντας τις τιμές που μας υποδεικνύεται στην εκφώνηση, παίρνουμε τα εξής αποτελέσματα:



Ουσιαστικά αλλάζει η περιοχή η οποία αντιγράφεται, μιας και αλλάζει το μήκος και το πλάτος, στα οποία αναφέρονται και οι τιμές που αλλάξαμε κατά την κλήση της μεθόδου `copyArea()`.

ε. Σταματώντας και ξαναξεκινώντας την εφαρμογή 3-4 φορές, έχουμε τα εξής:





Παρατηρούμε ότι σε κάθε εκτέλεση αλλάζει το χρώμα τόσο του φόντου, όσο και της γραμματοσειράς, η θέση εκτύπωσης, κτλ., σύμφωνα και με το αποτέλεσμα της συνάρτησης `Random()`.

3ο Σετ Ασκήσεων

1η Άσκηση

Ακολουθώντας τις οδηγίες της εκφώνησης, συντάξαμε το παρακάτω πρόγραμμα:

```
import javax.swing.*;
import javax.imageio.*;
import java.awt.image.BufferedImage;
import java.awt.event.*;
import java.awt.*;
import java.io.*;

public class Image_Application extends JFrame implements ActionListener{
    JMenuBar menu_bar;
    JMenu options_menu;
    JMenuItem reset_option;
    JMenuItem double_option;
    JPanel main_panel, image_panel;
    JPanel align_buttons, resize_buttons;
    JButton left_button, center_button, right_button;
    JLabel picture_label;
    int start_width, start_height;
    int temp_width, temp_height;
    int double_width, double_height;
    JLabel width_label, height_label;
    JTextField text_width, text_height;
    JButton resize_button;
    BufferedImage image;
    int type;

    String picture_path=new String("icon_01.jpg");

    public static void main (String args[]) {
        Image_Application image_application=new Image_Application();
    }

    public Image_Application() {
        super("Image Application");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        menu_bar=new JMenuBar();
        options_menu=new JMenu("Options");
        reset_option=new JMenuItem("Reset");
        reset_option.setActionCommand("reset");
        reset_option.addActionListener(this);
        double_option=new JMenuItem("Double");
        double_option.setActionCommand("double");
        double_option.addActionListener(this);
        options_menu.add(reset_option);
        options_menu.add(double_option);
    }
}
```

```

menu_bar.add(options_menu);
setJMenuBar(menu_bar);

main_panel=new JPanel();
main_panel.setLayout(new BorderLayout());
align_buttons=new JPanel(new GridLayout(1,3));
image_panel=new JPanel();
image_panel.setLayout(new FlowLayout(FlowLayout.CENTER));
resize_buttons=new JPanel(new GridLayout(3,2));
width_label=new JLabel("Width:");
text_width=new JTextField();
height_label=new JLabel("Height:");
text_height=new JTextField();
resize_button=new JButton("Resize");
resize_button.setActionCommand("resize");
resize_button.addActionListener(this);

resize_buttons.add(width_label);
resize_buttons.add(text_width);
resize_buttons.add(height_label);
resize_buttons.add(text_height);
resize_buttons.add(resize_button);

left_button=new JButton("Align Left");
left_button.setActionCommand("align_left");
left_button.addActionListener(this);
center_button=new JButton("Align Center");
center_button.setActionCommand("align_center");
center_button.addActionListener(this);
right_button=new JButton("Align Right");
right_button.setActionCommand("align_right");
right_button.addActionListener(this);

align_buttons.add(left_button);
align_buttons.add(center_button);
align_buttons.add(right_button);

try{
    image=ImageIO.read(new File(picture_path));
    type=image.getType()==0? BufferedImage.TYPE_INT_ARGB :
        image.getType();
    start_height=image.getHeight();
    start_width=image.getWidth();
    temp_height=start_height;
    temp_width=start_width;
} catch(IOException ioe){
    ioe.printStackTrace();
}

picture_label=new JLabel(new ImageIcon(image));
image_panel.add(picture_label);

```

```

        main_panel.add(aligned_buttons, BorderLayout.NORTH);
        main_panel.add(image_panel, BorderLayout.CENTER);
        main_panel.add(resize_buttons, BorderLayout.SOUTH);

        add(main_panel);

        setSize(550,675);
        setVisible(true);
    }

    public BufferedImage Resize_Image(BufferedImage original_image, int
        type, int new_width, int new_height) {
        temp_width=new_width;
        temp_height=new_height;
        BufferedImage resized_image=new BufferedImage(new_width,
            new_height,type);
        Graphics2D graph_image=resized_image.createGraphics();

        graph_image.drawImage(original_image, 0, 0, new_width, new_height,
            null);
        graph_image.dispose();

        return resized_image;
    }

    public BufferedImage Double_Image(BufferedImage original_image, int
        type, int new_width, int new_height) {
        double_width=2*new_width;
        double_height=2*new_height;
        temp_width=double_width;
        temp_height=double_height;
        BufferedImage doubled_image=new BufferedImage(double_width,
            double_height,type);
        Graphics2D graph_image=doubled_image.createGraphics();

        graph_image.drawImage(original_image,    0,    0,    double_width,
double_height, null);
        graph_image.dispose();

        return doubled_image;
    }

    public void actionPerformed(ActionEvent e) {
        if("align_left".equals(e.getActionCommand())) {
            image_panel.setLayout(new FlowLayout(FlowLayout.LEFT));
            image_panel.revalidate();
            validate();
            repaint();
        } else if("align_center".equals(e.getActionCommand())) {
            image_panel.setLayout(new FlowLayout(FlowLayout.CENTER));

```

```

        image_panel.revalidate();
        validate();
        repaint();
    } else if("align_right".equals(e.getActionCommand())) {
        image_panel.setLayout(new FlowLayout(FlowLayout.RIGHT));
        image_panel.revalidate();
        validate();
        repaint();
    } else if("resize".equals(e.getActionCommand())) {
        picture_label.removeAll();
        image_panel.removeAll();
        BufferedImage resized_image=Resize_Image(image, type,
            Integer.parseInt(text_width.getText()),
            Integer.parseInt(text_height.getText()));
        picture_label=new JLabel(new ImageIcon(resized_image));
        picture_label.setPreferredSize(new
            Dimension(Integer.parseInt(text_width.getText()),
            Integer.parseInt(text_height.getText())));
        picture_label.revalidate();
        image_panel.add(picture_label);
        image_panel.revalidate();
        validate();
        repaint();
    } else if("double".equals(e.getActionCommand())) {
        picture_label.removeAll();
        image_panel.removeAll();
        BufferedImage doubled_image=Double_Image(image, type,
            temp_width, temp_height);
        picture_label=new JLabel(new ImageIcon(doubled_image));
        picture_label.setPreferredSize(new Dimension(temp_width,
            temp_height));
        picture_label.revalidate();
        image_panel.add(picture_label);
        image_panel.revalidate();
        validate();
        repaint();
    } else if("reset".equals(e.getActionCommand())) {
        temp_width=start_width;
        temp_height=start_height;
        picture_label.removeAll();
        image_panel.removeAll();
        picture_label=new JLabel(new ImageIcon(image));
        picture_label.setPreferredSize(new Dimension(start_width,
            start_height));
        picture_label.revalidate();
        image_panel.setLayout(new FlowLayout(FlowLayout.CENTER));
        image_panel.add(picture_label);
        image_panel.revalidate();
        validate();
        repaint();
    }
}

```

}

}

Προς επιβεβαίωση της ορθής λειτουργίας του προγράμματος, παρατίθεται το ακόλουθο screenshot κατά την εκκίνησή του:



2η Άσκηση

Ακολουθώντας τις οδηγίες της εκφώνησης, γράψαμε την παρακάτω εφαρμογή:

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Puzzle_Game extends java.applet.Applet{
    JButton[] button_table=new JButton[9];
    String[] button_label={"1", "2", "3", "4", "5", "6", "7", "8"};
    boolean[] verification={false, false, false, false, false, false, false,
                                                                    false};

    int moves;
    JPanel game_panel=new JPanel();
    JPanel main_panel=new JPanel(new GridLayout(3,3, 10, 10));
    JPanel move_panel=new JPanel(new FlowLayout());
    JLabel move_text;
    JLabel game_moves;
    JButton new_game_button;

    public void init() {
        setSize(390, 450);
        setVisible(true);
        setLayout(new BorderLayout());
        new_game_button=new JButton("New Game");

        new_game_button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                moves=0;
                game_moves.setText(Integer.toString(moves));

                for(int i=0; i<8; i++) {
                    verification[i]=false;
                }

                for(int i=0; i<8; i++) {
                    Button_Loop: for(;;) {
                        int x=(int)(Math.random()*8);

                        if(verification[x]) {
                            continue Button_Loop;
                        } else {
                            button_table[i].setText(button_label[x]);
                            button_table[i].setBackground(new
                                Color(238,238, 224));
                            verification[x]=true;
                            break;
                        }
                    }
                }
            }
        });
    }
}
```

```

        }
    }

    boolean end_loop=false;

    for(int i=0;i<9;i++){
        for (int j=0;j<9;j++){
            if(i!=j && button_table[i].getText() ==
                button_table[j].getText()){
                button_table[i].setText("");
                button_table[i].setBackground(Color.black);
                end_loop=true;
                break;
            }

            if (end_loop==true){
                return;
            }
        }
    }
});

game_panel.add(new_game_button);
main_panel.setBackground(Color.black);

move_text=new JLabel("Moves:");
game_moves=new JLabel("0");
move_panel.add(move_text);
move_panel.add(game_moves);

add(game_panel, BorderLayout.NORTH);
add(main_panel, BorderLayout.CENTER);
add(move_panel, BorderLayout.SOUTH);

moves=0;
game_moves.setText(Integer.toString(moves));

for(int i=0;i<9;i++){
    button_table[i]=new JButton("");
    button_table[i].setFont(new Font("Tahoma", Font.BOLD, 35));
    button_table[i].setForeground(new Color(131, 139, 131));
    button_table[i].setBackground(Color.black);
    main_panel.add(button_table[i]);
}

for(int i=0;i<8;i++){
    Button_Loop: for(;;){
        int x=(int)(Math.random()*8);

```

```

        if(verification[x]) {
            continue Button_Loop;
        } else {
            button_table[i].setText(button_label[x]);
            button_table[i].setBackground(new Color(238,
            238,224));
            verification[x]=true;
            break;
        }
    }
}

for(int i=0;i<9;i++){
    button_table[i].addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ae) {
            if((ae.getSource()).equals(button_table[0])) {
                button_swap(0,1);
                button_swap(0,3);
            }

            if((ae.getSource()).equals(button_table[1])) {
                button_swap(1,0);
                button_swap(1,4);
                button_swap(1,2);
            }

            if((ae.getSource()).equals(button_table[2])) {
                button_swap(2,1);
                button_swap(2,5);
            }

            if((ae.getSource()).equals(button_table[3])) {
                button_swap(3,0);
                button_swap(3,4);
                button_swap(3,6);
            }

            if((ae.getSource()).equals(button_table[4])) {
                button_swap(4,1);
                button_swap(4,3);
                button_swap(4,5);
                button_swap(4,7);
            }

            if((ae.getSource()).equals(button_table[5])) {
                button_swap(5,2);
                button_swap(5,4);
                button_swap(5,8);
            }
        }
    });
}

```

```

        if((ae.getSource()).equals(button_table[6])) {
            button_swap(6,3);
            button_swap(6,7);
        }

        if((ae.getSource()).equals(button_table[7])) {
            button_swap(7,4);
            button_swap(7,6);
            button_swap(7,8);
        }

        if((ae.getSource()).equals(button_table[8])) {
            button_swap(8,7);
            button_swap(8,5);
        }
    }
});
}

public void button_swap(int x, int y){
    if((button_table[y].getText()).equals("")) {
        button_table[y].setText(button_table[x].getText());
        button_table[y].setBackground(new Color(238, 238, 224));
        button_table[x].setText("");
        button_table[x].setBackground(Color.black);
        moves++;
        game_moves.setText(Integer.toString(moves));
        check();
    }
}

public void check() {
    int button_number;
    boolean end_game=true;

    for(int i=0;i<8;i++){
        if(button_table[i].getText().equals("")) {
            return;
        }
        button_number=Integer.parseInt(button_table[i].getText());

        if(button_number!=i+1){
            end_game=false;
        }
    }

    if(end_game==true){
        JOptionPane.showMessageDialog(null, "Well Done!");
    }
}

```

```
}  
}
```

Προς επιβεβαίωση της ορθής λειτουργίας της εφαρμογής, παρατίθεται το παρακάτω screenshot από μια στιγμή της λειτουργίας του παιχνιδιού:

