

פרויקטים שהוכנו על-ידי משתתפי קורס מורים מובילים תש"ע

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

משחק סולמות וחבלים גרסת ג'אוה

פיתוח הפרויקט :

יהודה אבני: גרסת סי שרפ

אביטל (Evi) גרינולד: גרסת ג'אוה

דורית כהן: גרסת סי שרפ

המחלקות המרכיבות את הפרויקט

שם המחלקה	תיעוד
Game	מחלקה המדמה את המשחק.. מציגה את הלוח , הקובייה, ואת מיקום השחקנים על הלוח
Drawable	מחלקה מופשטת המייצגת עצם הניתן ליצור על פאנל
DrawPanel	מחלקה המכילה עצם מטיפוס Drawable וקואורדינטות x,y המהוות פינה שמאלית עליונה של העצם.
Board	לוח משחק עם מערך שחקנים
Player	מחלקת שחקן. לשחקן יש שם, צבע, מיקום, נקודות ומספר צעדים שעבר על הלוח.
Die	קוביית משחק

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

תרשימי UML למחלקות:

Game extends JFrame	
private Board board;	לוח משחק
private Player[] players;	מערך שחקנים
private Die die;	קוביית משחק
private DrawPanel boardPanel;	פאנל ללוח
private DrawPanel diePanel;	פאנל לקובייה
private DrawPanel player1Panel;	פאנל לשחקן 1
private DrawPanel player2Panel;	פאנל לשחקן 2
private JButton bRoll;	כפתור לזריקת קובייה
private JButton bMove;	כפתור לתזוזת שחקן על הלוח
private int index;	מספר מציין לשחקן
private Point newPos;	מיקום על הלוח
private int win=-1;	מציין למנצח
private boolean gameStarts=false;	האם משחק התחיל
public Game()	פעולה בונה משחק

Board extends Drawable	
static private Random rnd = new Random();	
final static int SIZE=9;	גודל הלוח
private Point[][] board;	לוח משחק ריבועי
private Player[] players;	מערך שחקנים
public Board(Player[] players)	פעולה יוצרת לוח עם שחקנים
public Point getCellValue(Point pos)	פעולה מחזירה מיקום בלוח
public Point getCellValue(int row, int col)	פעולה מחזירה מיקום בלוח
public void changeCelValue(int row, int col)	פעולה לשינוי ערך תא בלוח
public int getCellNum(Point pos)	פעולה המחזירה ערך מספרי למשבצת בלוח
public int getCellNum(int row, int col)	פעולה המחזירה ערך מספרי למשבצת בלוח
public void paint(Graphics g, int width, int height)	פעולה לציור לוח

Die extends Drawable	
static Random rnd = new Random();	
private Image img;	תמונה של קובייה
private int point;	ניקוד הקובייה
public Die()	יצירת קובייה
public int getNum()	איחזור ערך הקובייה
public void roll()	זריקת קובייה
public void paint(Graphics g, int width, int height)	ציור קובייה

Player extends Drawable	
private Point pos;	מיקום שחקן על הלוח
private String name;	שם שחקן
private int point;	מספר נקודות
private Color color;	צבע החייל של השחקן
private int steps;	מספר הצעדים שעבר על הלוח
public Player(String name, Color color)	פעולה יוצרת שחקן לפי הפרמטרים
public String getName()	מאחזרת שם שחקן
public Color getColor()	מאחזרת צבע שחקן
public int getPoint()	מאחזרת ניקוד שחקן
public Point getPosition()	מאחזרת מיקום שחקן על הלוח
public void setPoint(int point)	קובעת ניקוד חדש לשחקן
public void setPosition(Point pos)	קובעת מיקום חדש
public void increaseSteps()	מגדילה ב-1 את מספר הצעדים
public void paint(Graphics g, int width, int height)	פעולת ציור על פאנל

DrawPanel extends JPanel	
private Drawable d;	עצם הניתן לציור על פאנל
private int x;	פינה שמאלית עליונה - x
private int y;	פינה שמאלית עליונה - y
public DrawPanel(Drawable d, int x, int y, int width, int height)	פעולה היוצרת עצם הניתן לציור על פאנל לפי הפרמטרים
public void paint(Graphics g)	פעולת ציור העצם

abstract class Drawable	
public abstract void paint(Graphics g, int x, int y);	

שימוש במחלקות מיובאות

בלוח המשחק

import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.AffineTransform;
import java.awt.image.BufferedImage;
import java.util.Random;
import java.awt.Point;

בטיפוס DrawPanel :

```
import javax.swing.JPanel;  
import java.awt.Graphics;  
import java.awt.Dimension;
```

בקוביית המשחק:

```
import java.awt.Graphics;  
import java.awt.Graphics2D;  
import java.awt.Image;  
import java.awt.geom.AffineTransform;  
import java.io.File;  
import java.util.Random;
```

במחלקה Game:

```
import java.awt.Point;  
import java.awt.BorderLayout;  
import java.awt.Color;  
import java.awt.Dimension;  
import java.awt.Font;  
import java.awt.TextField;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
import javax.swing.BoxLayout;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JOptionPane;  
import javax.swing.JPanel;  
import javax.swing.border.LineBorder;
```

במחלקה DrawPanel

```
import javax.swing.JPanel;  
import java.awt.Graphics;  
import java.awt.Dimension;
```

המחלקות המרכיבות את המשחק – גרסת ג'אווה.

כתבה: אביטל (EVI) גרינולד, נעזרה בתלמיד : גבע קיפר, תלמיד כיתה י"ב בבי"ס חוגים, חיפה.

המחלקה Game : שלבי המשחק והמימשק הגרפי.

```
package ladderroap;
import java.awt.Point;
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.border.LineBorder;

import java.awt.Graphics2D;
/**
 * LadderRoap: contains Square Board, Die and 2 Players
 * @author EVI with Geva Kipper
 * @version 4/2/10 , update: 10/3/10
 */
public class Game extends JFrame
{
    private Board board;
    private Player[] players;
    private Die die;
    private DrawPanel boardPanel;
    private DrawPanel diePanel;
    private DrawPanel player1Panel;
    private DrawPanel player2Panel;
    private JButton bRoll;
    private JButton bMove;
    private int index; // Player's index
    private Point newPos; // position of Player
    private int win=-1; // no winner

    String name1 = JOptionPane.showInputDialog("1: הקלד שם שחקן");
    String name2 = JOptionPane.showInputDialog("2: הקלד שם שחקן");
/**
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

* create Game with 2 players, a Board and d die,
* button to roll the die and button to move the Player
*/
public Game()
{
    this.players = new Player[2]; // Array of Player with 2 players
    this.players[0] = new Player(name1,Color.blue);
    this.players[1] = new Player(name2,Color.red.darker());
    this.board = new Board(players); // מעבירים ללוח הפנייה למערך השחקנים
    this.die = new Die();
    this.getContentPane().setLayout(new BorderLayout()); // קובעים את הפריסה לפריים שלנו
    // בחרתי פריסה שמסדרת רכיבים לפי איזורים - מרכז, צפון, מזרח, מערב ודרום
    // יצירת הרכיבים והוספתם לפריים
    diePanel = new DrawPanel(die, 50,50); // יצירת הפאנל של הקוביה
    boardPanel = new DrawPanel(board, 500,500); // יצירת הפאנל של הלוח
    this.add(boardPanel, BorderLayout.CENTER); // הוספת הפאנל של הלוח במרכז הפריים
    JLabel header = new JLabel("סולמות וחבלים");
    header.setForeground(Color.magenta);
    header.setFont(new Font("Arial", Font.BOLD, 20));
    this.add(header, BorderLayout.NORTH);
    player1Panel = new DrawPanel(players[0], 60,120); // יצירת הפאנל של שחקן 1
    player2Panel = new DrawPanel(players[1], 60,120); // יצירת הפאנל של שחקן 2
    //player1Panel.setBorder(new LineBorder(Color.DARK_GRAY,4));
    /*
    JLabel label1 = new JLabel("1 שחקן"); // יוצר לייבל מנצה
    label1.setForeground(players[0].getColor()); // קובע את צבע הלייבל כצבע השחקן המנצה
    label1.setFont(new Font("Arial", Font.BOLD, 18)); // קובע פונט ללייבל
    this.add(label1, BorderLayout.SOUTH); // מוסיף לייבל במקום הנכון
    */
    bRoll = new JButton("הטל קוביה"); // יצירת כפתור הטל קוביה
    bRoll.setPreferredSize(new Dimension(100,50));
    bRoll.addActionListener(new ActionListener()
    { // מאזין להחיצה על הכפתור
        public void actionPerformed(ActionEvent arg0)
        {
            if (newPos!=null&&players[index].getPosition()!=newPos)
                finalPosition();
            index=(index+1)%players.length;
            die.roll();
            // update Players' points
            players[index].setPoint(players[index].getPoint()+die.getNum());
            player1Panel.repaint();
            player2Panel.repaint();
        }
    }
}

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
        diePanel.repaint();
        newPos=getNewPos(new Point(players[index].getPosition()),
                           die.getNum());
    }
    });

bMove = new JButton("זוז");
bMove.addActionListener(new ActionListener()
{ // מאזין לכפתור
    public void actionPerformed(ActionEvent arg0)
    {
        players[index].setPosition(newPos);
        players[index].increaseSteps();
        checkWin();
        if (newPos!=null&&board.getCellValue(newPos)!=newPos)
            newPos=board.getCellValue(newPos);
        boardPanel.repaint();
        player1Panel.repaint();
        player2Panel.repaint();
    }
});

JPanel p1 = new JPanel();
p1.setLayout(new BorderLayout(p1, BorderLayout.Y_AXIS));
p1.add(diePanel); p1.add(bRoll); p1.add(bMove);
this.add(p1, BorderLayout.EAST);

JPanel p2 = new JPanel();
p2.setLayout(new BorderLayout(p2, BorderLayout.X_AXIS));
p2.add(player1Panel); p2.add(player2Panel);
this.add(p2, BorderLayout.SOUTH);

// סיום יצירת הפריים
this.pack();
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
this.setVisible(true);
}
/-- private help methods

/**
 * @param position: Point, position of Player
 * @param num: number of steps to go
 * @return new position
```

```

*/
private Point getNewPos(Point position, int num)
{// מקבלת שחקן ומספר צעדים, מחזירה מיקום סופי שלו אחרי תזוזה
    for (int i=0;i<num;i++)
        oneStep(position);
    if (position.getX()>board.SIZE-1)
    { // out of Board, go back a one step
        position.setLocation(position.getX()-1,position.getY()-1);
    }
    return position;
}
/**
 * @param pos: Point, position of Player on Board
 * set new position after one step
 */
private void oneStep(Point pos)
{// מבצעת צעד אחד מהמיקום הנתון
    if (pos.getX()%2==0) {
        if (pos.getY()==board.SIZE-1)
            pos.setLocation(pos.getX()+1, board.SIZE-1);
        else pos.setLocation(pos.getX(), pos.getY()+1);
    }
    else
    {
        if (pos.getY()==0)
            pos.setLocation(pos.getX()+1, 0);
        else
            pos.setLocation(pos.getX(), pos.getY()-1);
    }
}
/**
 * check if there is a winner and stop the game
 */
private void checkWin()
{// בודקת אם שחקן ניצח, ואם כן עוצרת את המשחק
    for (int i=0;i<players.length;i++)
    {
        if (players[i].getPosition().getX()==board.SIZE-1 &&
            players[i].getPosition().getY()==(board.SIZE-1)*(board.SIZE%2))
        { //player won
            win=i;
            bRoll.setEnabled(false);
            bMove.setEnabled(false);
        }
    }
}

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.


```

JLabel l = new JLabel("שחקן " + (i+1) + " ניצח");
l.setForeground(players[i].getColor());
l.setFont(new Font("Arial", Font.BOLD, 18));
this.add(l, BorderLayout.SOUTH);
pack();
JOptionPane.showMessageDialog(this.getContentPane(), "שחקן " + (i+1) + " ניצח!");
    }
}
}

/**
 * Player moves to position according point and ladder or roap
 */
private void finalPosition()
{
    Point prev= newPos;
    newPos=board.getCellValue(newPos);
    while (prev!=newPos)
    {
        prev=newPos;
        newPos=board.getCellValue(newPos);
    }
    players[index].setPosition(newPos);
    boardPanel.repaint();
    checkWin();
}

public static void main(String[] args)
{
    new Game();
}
}

```

המחלקה Board : כוללת לוח משחק ריבועי + מערך שחקנים.
הלוח כולל פעולת paint לציון הלוח עם השחקנים.

```

package ladderroap;
import java.awt.BasicStroke;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.geom.AffineTransform;
import java.awt.image.BufferedImage;
import java.util.Random;
import java.awt.Point;
/**
 * Borad with Points on each cell. Point has coordinate of baord to move to

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

* 2 dimension array
* the board created in random mode
* @author EVI
* @version 25/2/10
*/
public class Board extends Drawable
{
    static private Random rnd = new Random();
    final static int SIZE=9; // 50
    private Point[][] board; // 2-dimension array of Points
    private Player[] players;
    /**
     * create Board
     */
    public Board(Player[] players)
    {
        this.board = new Point[SIZE][SIZE];
        this.players=players;
        int rndVal;
        int x,y;
        for (int row=0; row<SIZE; row++)
        {
            for (int col=0; col<SIZE; col++)
            {
                this.board[row][col] = new Point(row,col);
            }
        }
        // add ladders and roaps
        for (int k=1; k<=5; k++)
        {
            int row = rnd.nextInt(SIZE);
            int col = rnd.nextInt(SIZE);
            x = rnd.nextInt(SIZE);
            y = rnd.nextInt(SIZE);
            this.board[row][col] = new Point(x,y);
        } // end for k
        // dont change first and last cell
        this.board[0][0] = new Point(0,0);
        this.board[SIZE-1][SIZE-1] = new Point(SIZE-1,SIZE-1);
    }
    /**
     * @return String representing Board
     */
    public String toString()

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

{
    String str = "";
    for (int row=0; row<SIZE; row++)
    {
        for (int col=0; col<SIZE; col++)
        {
            str += String.format("[ %3.0f , %3.0f ] | ",
                this.board[row][col].getX(), this.board[row][col].getY());
        }
        str+= String.format("%n");
    }
    return str;
}
/**
 * @param pos: Point . position of Board's cell
 * @return value of cell in pos
 */
public Point getCellValue(Point pos)
{
    int row = (int) pos.getX();
    int col = (int) pos.getY();
    //System.out.println("inside Board: " + this.board[row][col]);
    return this.board[row][col];
}
/**
 * @param row:
 * @param col:
 * @return value of cell in pos
 */
public Point getCellValue(int row, int col)
{
    return this.board[row][col];
}
public void displayCellValue(Point pos)
{
    Point cell = this.getCellValue(pos);
    System.out.println("[ " + (int)cell.getX() + " , " + (int)cell.getY() + " ]");
}
/**
 * change value in cell row,col of Board randomly
 * @param row: integer
 * @param col: integer
 * assumption: row and col are vccorrect indexes

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
*/
public void changeCelValue(int row, int col)
{
    int rndVal;
    int x,y;
    x = rnd.nextInt(SIZE);
    y = rnd.nextInt(SIZE);
    Point newPos = new Point(x,y);
    this.board[row][col] = newPos;
}
/**
 * @param row:integer
 * @param col:integer
 * @return number of cell on Baord according to row and column
 */
public int getCellNum(Point pos)
{
    int row = (int) pos.getX();
    int col = (int) pos.getY();
    int num=0;
    if (row%2==0)
        num = row*SIZE+col+1;
    else // row%2==1
        num =row*SIZE + SIZE-col;
    return num;
}
/**
 * @param row:integer
 * @param col:integer
 * @return number of cell on Baord according to row and column
 */
public int getCellNum(int row, int col)
{
    int num=0;
    if (row%2==0)
        num = row*SIZE+col+1;
    else // row%2==1
        num =row*SIZE + SIZE-col;
    return num;
}
/**
 * This method paints the board on the graphics object g,
 * scaled to fit the given width and height dimensions
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

*/
public void paint(Graphics g, int width, int height)
{
    width=Math.min(width,height);
    int w=width/this.SIZE;
    /*BufferedImage img = new BufferedImage(width,width,
        BufferedImage.TYPE_INT_ARGB);
    Graphics g = img.getGraphics();*/
    //Draw the board
    g.drawRect(0, 0, SIZE*w, SIZE*w);
    g.setColor(Color.gray);
    g.fillRect(0, 0, SIZE*w, SIZE*w);
    g.setColor(Color.black);
    g.setFont(new Font("Arial", Font.BOLD, 18));
    int num; // value of cell
    for (int row=0;row<SIZE;row++)
    {
        for (int col=0;col<SIZE;col++)
        {
            g.drawRect(row*w, col*w, w, w); //מצייר ריבוע (row,col)
            if (row%2==0)
                num = row*SIZE+col+1;
            else // row%2==1
                num =row*SIZE + SIZE-col;
            g.drawString(""+num, row*w+w/2-5, col*w+w/2-5);
        }
    }

    //מצייר סולמות וחבלים
    Graphics2D g2 = (Graphics2D) g;
    g2.setStroke(new BasicStroke(2));
    g.setColor(Color.black);
    g.setFont(new Font("Arial", Font.BOLD, 18));
    int num1,num2;
    for (int row=0;row<board.length;row++)
    {
        for (int col=0;col<board.length;col++)
        {
            if (!board[row][col].equals(new Point(row,col)))
            { //חבל או סולם יוצא
                //מצייר קו בין הריבוע הזה לריבוע עליו הוא מצביע
                num1 = this.getCellNum(row, col);
                num2 = this.getCellNum(this.getCellValue(row,col));
            }
        }
    }
}

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
        if (num2>num1)
        {
            g2.setColor(Color.GREEN);
            g2.setStroke(new BasicStroke(3));
        }
        else
        {
            g2.setColor(Color.WHITE);
            g2.setStroke(new BasicStroke(2));
        }
        g.drawLine(row*w+w/2, col*w+w/2,
            (int)board[row][col].getX()*w+w/2,
            (int)board[row][col].getY()*w+w/2);
        g.fillOval(row*w+w/4, col*w+w/4, w/3, w/3); // מצייר עיגול במרכז הריבוע הזה
    }
}

//2 מצייר את השחקנים בהנחה שיש בדיוק
// draw first player
g.setColor(players[0].getColor());
g.fillOval((int)players[0].getPosition().getX()*w+w/4,
    (int)players[0].getPosition().getY()*w, w/2, w/2);
// draw second player
g.setColor(players[1].getColor());
g.fillOval((int)players[1].getPosition().getX()*w+w/4,
    (int)players[1].getPosition().getY()*w+w/2, w/2, w/2);
}
}
```

המחלקה : Player – מחלקת שחקן.

לכל שחקן שמור מיקומו על לוח המשחק. שמו, הניקוד שלו – ניקוד מצטבר לפי זריקת קובייה, מספר הצעדים שעשה, צבע.

המחלקה כוללת את הפעולה paint לתצוגת נתוני שחקן.

```
package ladderroap;
import java.awt.*;
/**
 * Player: has points, position of Board, Color
 * @author EVI
 * @version 3/2/10
 */
public class Player extends Drawable
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

{
private Point pos; // position on Board
private String name; // Player's name
private int point; // points of Player
private Color color; // color of Player's soldier
private int steps; // number of steps to finish
/**
 * create Player with name, color and position at start and 0 points
 * @param name: String
 * @param color: Color
 */
public Player(String name, Color color)
{
    this.name = name;
    this.point = 0;
    this.pos = new Point(0,0);
    this.steps = 0;
    this.color = color;
}
/**
 * @return Player's name
 */
public String getName()
{
    return this.name;
}
/**
 * @return Color of layer's soldier
 */
public Color getColor()
{
    return this.color;
}
/**
 * @return Player's points
 */
public int getPoint()
{
    return this.point;
}
/**
 * @return Player's position on Board
 */
public Point getPosition()
{
    return this.pos;
}
/**
 * set points of Player to be point
 * @param point

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```

*/
public void setPoint(int point)
{
    this.point = point;
}
/**
 * set position on Board to be index
 * @param pos: integer
 */
public void setPosition(Point pos)
{
    this.pos = pos;
}
/**
 * increase number of steps by 1
 */
public void increaseSteps()
{
    this.steps++;
}
/**
 * @return String representing Player's name
 */
public String toString()
{
    String str="Player:\n";
    //str+= String.format("%10s %3d %3d %10s", this.name, this.point, this.index, this.color);
    str+= String.format("%10s has %3d points , ", this.name,this.point);
    int x = (int) this.pos.getX();
    int y = (int) this.pos.getY();
    str+= String.format(" in position: ( %3d , %3d ) ", x,y);
    return str;
}
public void paint(Graphics g, int width, int height)
{
    Graphics2D g2 = (Graphics2D) g;
    g2.setStroke(new BasicStroke(3));
    g2.clearRect(0, 0, width, height);
    width=Math.min(width,height);
    g2.setColor(Color.lightGray);
    g2.drawRect(0, 0, width, height);
    g2.fillRect(0, 0, width, height);
    g2.setFont(new Font("Arial", Font.PLAIN, 16));
    g2.setColor(this.getColor());
    g2.setFont(new Font("Arial", Font.BOLD, 16));
    g2.drawString("שחקן שם: "+this.name, 20, 20);
    g2.setFont(new Font("Arial", Font.PLAIN, 16));
    g2.drawString("ניקוד: "+this.point, 10, 40);
}

```

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.


```
        g.drawString("צעדים מספר: "+this.steps,10,60);
    }
} // end class
//=====
```

[המחלקה Die](#) : – קוביית משחק, אשר מציירת קובייה לפי תמונה. בעלת תכונה פרטית אחת – ניקוד על הקובייה.

```
package ladderroap;
/**
 * Die: game diw contains number 1-6, what point shows the die?
 * @author Evi
 * @version 29/8/08
 */
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.geom.AffineTransform;
import java.io.File;
import java.util.Random;
import javax.swing.ImageIcon;
public class Die extends Drawable
{
    static Random rnd = new Random();
    Image img;
    // properties
    private int point;

    /**
     * create Die with point on top (1-6)
     */
    public Die()
    {
        roll();
    }

    /**
     * @return number on top
     */
    public int getNum()
    {
        return this.point;
    }

    /**
     * @return a String representing Die
     */
    public String toString()
    {
        String str = "Die: " + this.point;
        return str;
    }
}
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
/**
 * roll Die
 */
public void roll()
{
    this.point = rnd.nextInt(6)+1;
    img= new ImageIcon(point+".JPG").getImage();
}

/**
 * Draws the die on the Graphics object g
 */
int w=75;
public void paint(Graphics g, int width, int height) {
    g.drawImage(img, 0, 0, w, w, null);
}
}

//=====
```

המחלקה `DrawPanel` : מחלקה לציור פאנל על פי עצם `Drawable` ומיקום (x,y) – פינה שמאלית עליונה של הפאנל.

```
package ladderroap;
import javax.swing.JPanel;
import java.awt.Graphics;
import java.awt.Dimension;
/**
 * DrawPanel : has Panel to paint and (x,y) - left corner of Panel
 * @author Geva Kipper, update by Evi Gruenwald
 * @version 8/3/10
 */
public class DrawPanel extends JPanel
{
    private Drawable d;
    private int x;
    private int y;
    /**
     * @param d: DrawPanel
     * @param x: x-coordinate of left up corner
     * @param y: y-coordinate of left up corner
     * @param width: Panel's with
     * @param height: Panel's height
     */
    public DrawPanel(Drawable d, int x, int y, int width, int height)
    {
        this.d=d;
        this.x=x; this.y=y;
        setPreferredSize(new Dimension(width,height));
    }
}
```

ניתן להשתמש בחומרים לצורך הוראה בלבד.
לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

```
}
/**
 * @param d: DrawPanel
 * @param x: x-coordinate of left up corner
 * @param y: y-coordinate of left up corner
 */
public DrawPanel(Drawable d, int width, int height)
{
    this(d,0,0, width, height);
}
/**
 * @param g: Graphics
 * paint the Pannel and allow repainting it
 */
public void paint(Graphics g)
{
    d.paint(g, getWidth(), getHeight());
}
}

//=====
```

מחלקה מופשטת: [Drawable](#) – מייצגת כעצם הניתן לציור על פאנל.

```
package ladderroap;
import java.awt.Graphics;
/**
 * @author Geva Kipper
 * @version 8/3/10
 */
public abstract class Drawable
{
    public abstract void paint(Graphics g, int x, int y);
}
```