

מערכת גרפית או GUI - Graphic User Interface

כל מערכת הפעלה אשר תומכת בסביבת GUI למשתמש עוקבת כל העת אחרי אירועים (events) שמתרחשים (הפעלת העכבר, הפעלת המקלדת...), ומדווחת בהתמדה לתכניות שרצות על האירועים. כל תכנית שמקבלת דיווח על האירועים שהתרחשו מחליטה בעצמה על התגובה (אם בכלל). ניתן להבחין באובייקטים שמסוגלים "ליצור" events כדוגמת: List, Choices, Buttons ואחרים... כל אחד מאובייקטים אלה ייקרא בשם event source. לכל אחד מהם ניתן יהיה לקבוע כי אובייקט אחר "יאזין" ל-events שנוצרים על ידו. אובייקט ש"מאזין" ל-events (האובייקטים שנוצרים על ידי האובייקט אחר) ייקרא בשם listener object. הוא חייב להשתייך למחלקה שמיישמת בהגדרתה את ה-interface המתאים לסוג ה-events שלהם הוא מאזין. כל המחלקות שמתארות events יורשות מן המחלקה java.util.EventObject. אובייקטים שמייצגים מרכיבי GUI מסוגלים בדרך כלל לייצר events מסוגים שונים. כך למשל, אובייקט מטיפוס Button מסוגל לייצור events מטיפוס ActionEvent, ואובייקט מטיפוס Window מסוגל לייצור WindowEvent.

יש חמש ספריות של מחלקות אשר נקראות יחדיו בשם JFC (Java Foundation Classes – JFC). כל ספריה כוללת בתוכה מחלקות שמשמשות למטרה מסוימת.

- AWT – ספריית מחלקות שמאפשרות יצירת GUI בסיסי.
- SWING – ספריית מחלקות שמאפשרות פיתוח GUI מתקדם שחזותו איננה תלויה במחשב שבו התכנית רצה. ספריה זו מהווה את עיקר ה-JFC.
- Java 2D – מחלקות שמשמשות ליצירת גרפיקה ברמה גבוהה בשני ממדים.

אנו נתמקד ב-SWING.

SWING היא ספריה של מחלקות שמספקות מרכיבי GUI ושירותים אחרים. כיוון שמחלקות ה-SWING כתובות באמצעות JAVA בלבד (pure 100% JAVA) קיימת אי-תלות בין האופן שבו נראים מרכיבי ה-GUI לבין המחשב שבו התכנית רצה. בעוד שמרכיבי ה-GUI ששייכים ל-AWT עשויים להיראות באופן שונה במחשבים שונים רכיבי ה-GUI ששייכים ל-SWING מוצגים באופן זהה בכל מחשב ומחשב. האפשרויות ביצירת GUI באמצעות מחלקות ה-SWING הרבה יותר מגוונות מהאפשרויות שקיימות כאשר משתמשים ב-AWT.

רכיבי יסוד של SWING:

רכיב Component – רכיב ממשק משתמש בסיסי, כגון: תווית, כפתור, רשימה, תיבת גלילה, רכיב טקסט. רכיב כולל פעולות גרפיקה בסיסיות, כגון: קביעת מיקום, קביעת גודלת הצגה/הסתרה, קביעת הגופן, טיפול באירועים. מחלקת Component יורשת מ-Object.

רכיבים שכיחים:

תווית JLabel – שורת טקסט קבוע.

כפתור JButton – כפתור לחיצה.

תיבת סימון JCheckBox – תיבה עם אפשרות לסימון או לביטול סימון

בחירה, רשימה JComboBox – תיבה להצגת רשימת פרטיים לבחירת המשתמש.

רכיב טקסט JTextField, JTextArea – רכיבים להקלדת טקסט בשורה אחת או יותר.

פעולות על רכיבים:

הפעולות הנפוצות היותר:

get/setBounds() – קבלת/קביעת מיקום וגודל הרכיב.

get/setFont() – קבלת/קביעת גופן של הרכיב.

get/setSize() – קבלת/קביעת גודל הרכיב.

repaint/update() – עדכון תצוגת הרכיב, גורם לקריאת ה-paint.

paint() – צביעת הרכיב לחלון.

setVisible() – קביעת מצב הרכיב כנראה.

טיפול באירועים:

addMouseListener() – רישום מאזין לאירועי לחיצה על העכבר.

addKeyListener() – רישום מאזין לאירועי לחיצה על מקשי המקלדת.

מכיל Container – רכיב המכיל רכיבים אחרים כגון:

מסגרת (Frame), תיבת דו-שיח, פנל (Panel), JApplet. הם כוללים פונקציונאליות לסידור הרכיבים המוכלים בהם. מחלקת Container יורשת

מ-Component. מיכל יכול להכיל רכיבים, ומכוון שמחלקת המיכל יורשת מ-Component היא יכולה להכיל באופן רקורסיבי גם מיכלים.

המיכלים השימושיים ביותר:

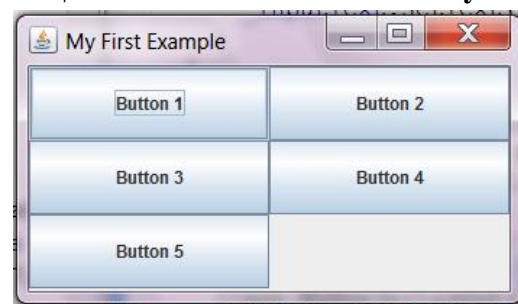
מסגרת (JFrame) – מיכל המציג את חלון המסגרת של התוכנית, עם כפתורי הקטנה, הגדלה וסגירה. כמו כן הוא מכיל את התפריטים.

פנל (JPanel) – מיכל המייצג את שטח מסוים בחלון ומכיל רכיבים גרפיים. פנל בדרך כלל מוכל במסגרת ומשמש לאיגוד של קבוצת רכיבים.

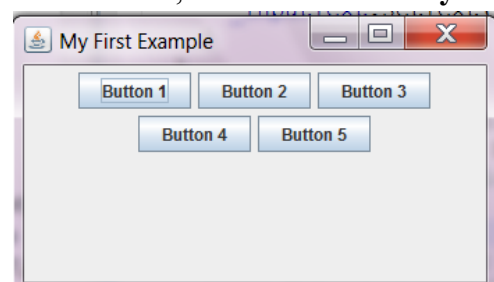
JApplet – מיכל המייצג יישום הניתן לשיפוף בדף HTML ולהרצה מתוך דפדפן.

סידור הרכיבים במיכל Component Layout-

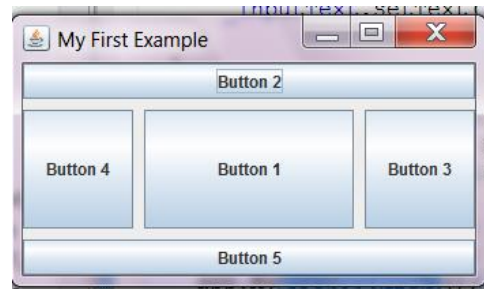
GridLayout – סידור הרכיבים במספר קבוע של שורות ועמודות:



FlowLayout – סידור זרימה, הרכיבים מסודרים באופן עוקב משמאל לימין ומלמעלה למטה:



BorderLayout – סידור הרכיבים בגבולות המכיל: צפון מערב, דרום, מזרח ואמצע:



מודל האירועים (**Event Model**) – מגדיר את המנגנון שבו התוכנית מגיבה לפעולות המשתמש, כגון: לחיצה על מקשי העכבר, הקשה על המקלדת וכו'.

קישור לאתר של JAVA

<http://www.java2s.com/Tutorial/Java/CatalogJava.htm>