

דוגמא נוספת – רשימת משימות

```

public class TaskList1 {
    private static final Display display =
        Display.getDefault();
    private static Shell shell;

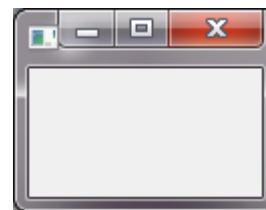
    public static void main(String[] args) {
        TaskList1 taskList = new TaskList1();
        taskList.createShell();
        taskList.runApplication();
    }

    private void createShell() {
        shell = new Shell(display);
        shell.setText("My Tasks");
    }

    private void runApplication() {
        shell.pack();
        shell.open();
        while (!shell.isDisposed()) {
            if (!display.readAndDispatch())
                display.sleep();
        }
        display.dispose();
    }
}

```

- נכתוב אפליקציית SWT פשוטה המאפשרת להזין משימות בשדה טקסט, ללחוץ על כפתור ולהוסיף את הטקסט שהוזן לרשימה.
- נתחיל מבניית שלד האפליקציה



הוספת Widgets

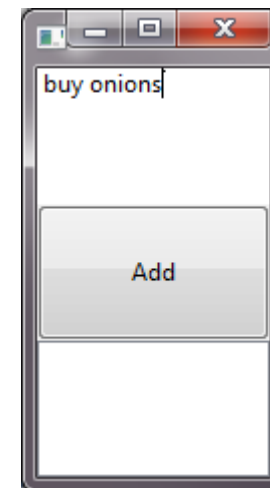
- נוסף Layout בסיסי ל-Shell, וכן שדה טקסט, כפתור ורשימה
- כרגע אין מאזינים לאף פעולה

```
private void createShell() {
    shell = new Shell(display);
    shell.setText("My Tasks");
    shell.setLayout(new FillLayout(SWT.VERTICAL));
```

```
//a text field to enter a task
Text input = new Text(shell, SWT.LEFT);

//a button to add a task to the list
Button add = new Button(shell, SWT.PUSH);
add.setText("Add");

//the list
List list = new List(shell, SWT.BORDER);
}
```



המחלקה SWT

- מוגדרת ב `org.eclipse.swt.SWT`
- אוסף של קבועים:
 - אירועים – `MouseDown`, `FocusIn`, `Close`, `Activate` ...
 - צבעים – `COLOR_BLUE`, `COLOR_BLACK` ...
 - תווים – `ESC`, `DEL`, `CR` ...
 - אירוע מקשים – `END`, `ARROW_DOWN` ...
 - עיצובים
- ניתן להוסיף מס' קבועים ע"י שימוש באופרטור | (bitwise OR)
 - `SWT.V_SCROLL | SWT.H_SCROLL | SWT.BORDER`

הוספת פונקציונליות

```
//a text field to enter a task
final Text input = new Text(shell, SWT.LEFT);

//a button to add a task to the list
Button add = new Button(shell, SWT.PUSH);
add.setText("Add");

// the list
final List list = new List(shell, SWT.BORDER);

// the action to perform when pressing the button
add.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        String text = input.getText();
        //add the task to the list, if the input text is not empty
        if (text != null && text.length() > 0) {
            list.add(text);
            input.setText("");
        }
    }
});
```

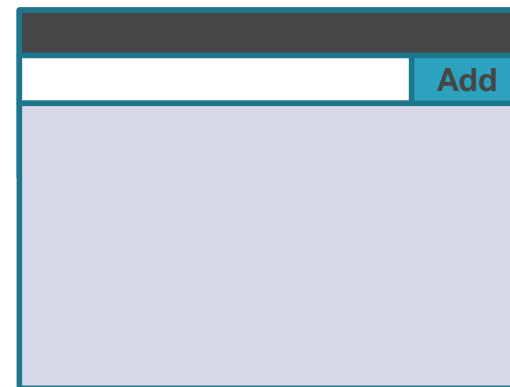
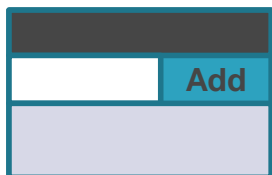
- כעת, נוסיף את הפעולה של לחיצה על כפתור – תוך שימוש במחלקה אנונימית.
- הכפתור צריך להכיר את שדה הטקסט ואת הרשימה, לכן נשנה אותם ל-final
- עכשיו זה עובד!

נטפל בעיצוב החלון

- תחילה, נתכנן איך היינו רוצים שהחלון ייראה



- וכיצד הוא ישתנה בשינוי גודל החלון



עיצוב החלון

```

shell = new Shell(display);
shell.setText("My Tasks");
shell.setLayout(new GridLayout(2, false));

// a text field to enter a task
final Text input = new Text(shell, SWT.LEFT);
input.setLayoutData(new GridData(
    GridData.FILL_HORIZONTAL));

// a button to add a task to the list
Button add = new Button(shell, SWT.FLAT);
add.setText("Add");

// the list
final List list = new List(shell, SWT.BORDER);
GridData listGridData = new GridData(GridData.FILL_BOTH);
listGridData.horizontalSpan = 2;
listGridData.widthHint = 300;
listGridData.heightHint = 300;
list.setLayoutData(listGridData);

```

שני טורים ברוחב
לא אחיד

• נשתמש ב-GridLayout

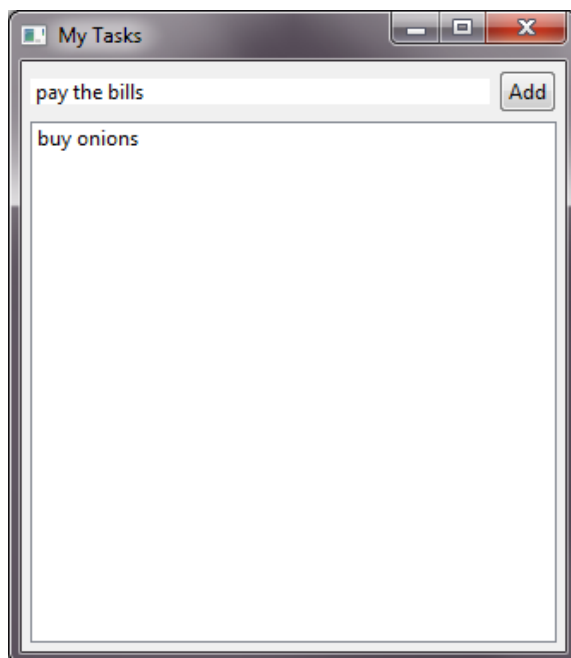
- מוסיף אלמנטים לגריד משמאל לימין ומלמעלה למטה לפי הסדר
- מאפשר הוספת GridData שמציין איך כל Widget יתנהג בגריד

שדה הטקסט נמתח לרוחב

הרשימה נמתחת לרוחב ולאורך, תופסת שני טורים, וגודלה המומלץ 300x300 פיקסלים

עיצוב החלון

- עכשיו האפליקציה שלנו נראית כך



- עוד תוספות?
 - סימון משימה כ"הושלמה"
 - הוספת תאריך יצירת המשימה
 - הוספת תאריך יעד לביצוע המשימה
 - עריכת משימה
 - מחיקת משימה
 - ...

תוספת - מחיקת משימות מהרשימה

- נוסף KeyListener לרשימה שיאזין ללחיצה על delete
- מ-Widget הרשימה ניתן לקבל את האינדקס של הפריט הנבחר ולמחוק את הפריט

```
// the action to perform when DELETE is pressed on the list
list.addKeyListener(new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        if (e.character == SWT.DEL) {
            int selectionIndex = list.getSelectionIndex();
            //if a list item is selected, delete it from the list
            if (selectionIndex >= 0)
                list.remove(selectionIndex);
        }
    }
});
```