הערות לשיעורים:

שיעור 7 – דפים 13, 14, 15:

Applets:

כשה-applet נטען אזי:

1. נוצר מופע של המחלקה השולט ב-applet.
2. ה-applet מאתחל את עצמו (initialize).
3. ה-applet מתחיל לרוץ.

כשהמשתמש עוזב את הדף ל-applet יש אפשרות לעצור (stop) את עצמו. כשהמשתמש חוזר לדף, ה-applet יכול להתחיל (start) את עצמו מחדש (כנ"ל כאשר המשתמש סוגר את החלון או עושה minimize). כשהמשתמש סוגר את החלון, ה-applet יכול לעצור (stop) את עצמו ולעשות final cleanup (destroy) לפני שהדפדפן נסגר.

מתודות ב-applet:

1. ***public void init()*** אתחול ה-applet בכל פעם שהיא עולה (מתבצע רק בהעלאת העמוד!). ה-init שימושי לאתחול חד פעמי שלא לוקח הרבה זמן. באופן כללי, ה-init מכיל את הקוד שבד"כ מופיע ב-constructor.
2. ***public void start()*** התחלת הביצוע של ה-applet כשהיא עולה או כשהמשתמש מבקר שוב בדף. כל applet שעושה משהו אחרי האתחול חייב לדרוס את start. start מבצע את פעולות ה-applet או יוצר thread אחד או יותר לביצוע העבודה.
3. ***public void stop()*** הפסקת הביצוע של ה-applet. רוב ה-applet שדורסות את start, דורסות גם את stop. stop צריך להשעות את ביצוע ה-applet כך שהוא לא יגזול משאבים מהמע' כשהמשתמש לא צופה בדף של ה-applet (למשל, עצירת מוסיקת הרקע כשלא מסתכלים בדף).
4. ***public void destroy()*** ביצוע final cleanup כהכנה ל-unloading של ה-applet. בד"כ לא צריך לדרוס את destroy שכן ב-stop (שנקראת לפני destory) עושים את כל מה שנחוץ בכדי לעצור את ביצוע ה-applet. אבל, ניתן להשתמש ב-destroy אם צריך לשחרר משאבים נוספים.
5. ***public void paint(Graphics g)*** המתודה הבסיסית לתצוגה. יישומונים רבים ממשים את המתודה הזו כדי לצייר את התצוגה של היישומון בתוך הדפדפן.
6. ***public void update(Graphics g)*** מתודה שניתן להשתמש בה ביחד עם paint עמ"נ לשפר את ביצועי הציור/תצוגה.

import java.net.*; - מכיל מחלקות שקשורות לתקשורת (בין היתר מחלקה של URL).

```
URL url = new URL(…);
… = url.getContent;
```

ב-URL משתמשים לתמונות ולצלילים (ב-Applet).

```
import java.applet.*;

class myApp extends Applet { … }
```

הערות לדף – הפיכת יישום ל-Applet:

1. g.drawString(string, x, y), כאשר (0, 0) זוהי הפינה השמאלית העליונה.
3. צריך להבטיח שה-applet יוכל להיעצר (ברגע שהחלון לא פעיל) – ע"י listener ל-window ודריכת window closing.
5. בד"כ כל מה שיש ב-()main זה יצירת Frame (קריאה ל-constructor), שינוי הגודל והצגתו ← לא צריך main כשיוצרים applet.
7. ב-Applet הדף גורם להעלאת היישומון. אין קריאה ל-constructor. במקום ה-constructor יש מתודה ()init שנקראת ברגע שהדף עולה. אחרי הקריאה ל-()init יש קריאה ל-()start.
   ה-()init נקרא רק פעם אחת – אתחול ראשוני.
   ()start נקרא כל פעם שהחלון גדל או מקבל focus.
   ()stop – כשהחלון קטן.
   ()destory – לאחר ()stop. מבצע final cleanup.
   כל applet חייבת לממש לפחות אחת מהמתודות הבאות: init, start, paint.
8. ה-default layout של Frame זה BorderLayout. של Panel and Applet זה Flow Layout. לכן, לא כדאי להסתמך על ה-default layout אלא להגדיר בתוכנית את ה-layout הרצוי.

תגי ה-applet ב-HTML:

```
<body>
<applet code=___(1) width=___ Height=____>
</applet>
</body>
```

(1) – קובץ ה-class (עם הסיומת class.)

הערה: אם כוללים applet פעמיים בעמוד, אזי הדפדפן מעלה את קובץ ה-class פעם אחת ויוצר 2 מופעים של המחלקה.

כשה-applet צריך להעלות מידע מקובץ שידוע עבורו ה-URL היחסי שלו ביחס ל-applet בד"כ משתמשים ב-
1. getCodeBase(): מחזיר URL שמגדיר את הספרייה שממנה נטענו קבצי ה-class של ה-applet.
2. getDocumentBase(): מחזיר URL שמגדיר את הספרייה שממנה נטען קובץ ה-HTML של ה-applet.

הערה: מטעמי בטיחות, רוב הדפדפנים לא מאפשרים להשתמש ב-".." כדי להגיע לספריות מעל הספרייה שמכילה את ה-codeBase/documentBase.

הגבלות דפדפנים על applets:
- applet לא יכול להעלות ספריות או להגדיר native methods.
- applet לא יכול לקרוא/לכתוב קבצים שמצויים על הלקוח שמריץ אותו.
- applet לא יכול ליצור network connections, חוץ מאשר ללקוח שממנו הוא בא.
- applet לא יכול להתחיל תוכניות על הלקוח שמריץ אותו.
- applet לא יכול לקרוא חלק מה-system properties.

הערה: הצגת מידע ב-status line של הדפדפן: showStatus(String).

ע"י <PARAM> ו-getParameter() ניתן להעביר נתונים מדף ה-HTML ל-applet, למשל:
```
<applet code=_______(only relative URL)>
<param name=title value=Raz>
</applet>
…
String windowTitle = getParameter("title");
```

הצגת הודעה במקרה של בעיות בהעלאת ה-applet:
```
<applet code = ___ ALT= "Error loading applet"> // for browsers that support Java
Java Not Supported // for non-enabled java browsers. Java supported browsers will ignore this line.
</applet>
```

http://java.sun.com/docs/books/tutorial/applet/practical/index.html

דפים 14, 15:

Threads:
Thread (או execution context או lightweight process) הוא קטע סדרתי בתוכנית של זרימת בקרת. משתמשים ב-threads עמ"נ לבודד משימות.
Thread דומה לתוכנית סדרתית, אולם thread כשלעצמו אינו תוכנית. הוא אינו יכול לרוץ בפני עצמו – הוא רץ בתוך תוכנית.

יש 2 שיטות ליצירת Thread:
1. *ירושה מ-Thread ודריסת run():* public class myThread extends Thread ואז צריך לדרוס את run(). בשיטה זו לא ניתן לרשת יותר!
2. *מימוש מנשק Runnable:* public class myThread implements Runnable – בשיטה זו צריך לממש את run().

מתי להשתמש במה? ***"חוק אצבע":*** אם המחלקה צריכה לרשת ממחלקה אחרת צריך להשתמש ב-Runnable.

p.start(); - מקצה את כל משאבי המע' הנחוצים ל-thread, מתזמנת את ריצת ה-thread ומפעילה את run() כשמסיים – כל המשאבים משתחררים.

מתודת ה-run() אומרת ל-thread מה לעשות. ניתן לכלול בתוכה כל מה שניתן לעשות בפקודות ב-Java.

Thread עובר למצב not runnable אם:
1. מתודת ה-Sleep שלו הופעלה.
2. ה-Thread קורא ל-wait.
3. ה-Thread חסום על פעולת ק/פ.

מחלקת Thread מממשת generic thread שלא עושה כלום (המימוש של run() הוא ריק).