

תרגיל בית מס. 5 – JScrollPane

את ההגשות יש לשלוח, בזוגות, לאתר הקורס עד 22 ביוני 2022. אנא הקדישו זמן לקריאת ההוראות שלהלן: התרגיל לא פשוט ותחסכו הרבה טעויות אם תקראו את הכתוב כאן בעיון.

בתרגיל זה נממש מחלקה המייצגת Widget (כמו JPanel, JButton) ומחלקות אחרות שראינו). מחלקה זו עוקבת אחרי תבנית העיצוב של **עיטור** (Decoration) - היא אינה חלון בפני עצמה, אבל היא מוסיפה יכולות חלונאיות לחלונות אחרים אותם היא "מקיפה". בפרט, היא מוסיפה את היכולת **לגלול** (scroll) תוכן של חלון אם הוא חורג מהגודל הפיזי של החלון בו הוא מוצג.

קיימת ב-Swing מחלקה הממלאת את התפקיד הזה ושמה JScrollPane. מובן שאין לנצל אותה כחלק מפתרון התרגיל הזה, אך בהחלט מומלץ לנסות לכתוב קוד שמשתמש בה על מנת ללמוד את תכונותיה, תכונות שנרצה להעניק גם ל-MyScrollPane.

1. הגדרת הדרישות:

MyScrollPane מקבלת (כארגומנט לבנאי) חלון אחד אותו היא אמורה "להקיף" – התוכן שלו הוא התוכן שיש לגלול. חלון מוכל זה עשוי להיות גדול יותר מ-MyScrollPane או אפילו מהמסך הפיזי עצמו: תפקידו של MyScrollPane הוא להוסיף על חלון זה, אם צריך, **זחיתים** (scroll bar) – אחד לגלילה אנכית ואחד לגלילה אופקית, כדי לאפשר צפייה בכל תוכן החלון גם אם הוא גדול מחלון התצוגה.

אין מגבלה על התוכן, והוא יכול לכלול כל דבר שניתן להצגה בחלון – טקסט, Widgets אחרים, כולל MyScrollPane אחרים. שתי המגבלות היחידות הן:

1. כל התוכן צריך להיות ארוז בחלון יחיד – זה שנמסר כפרמטר ל-MyScrollPane.
2. החלון הזה גדול מספיק כדי להציג את התוכן כולו.

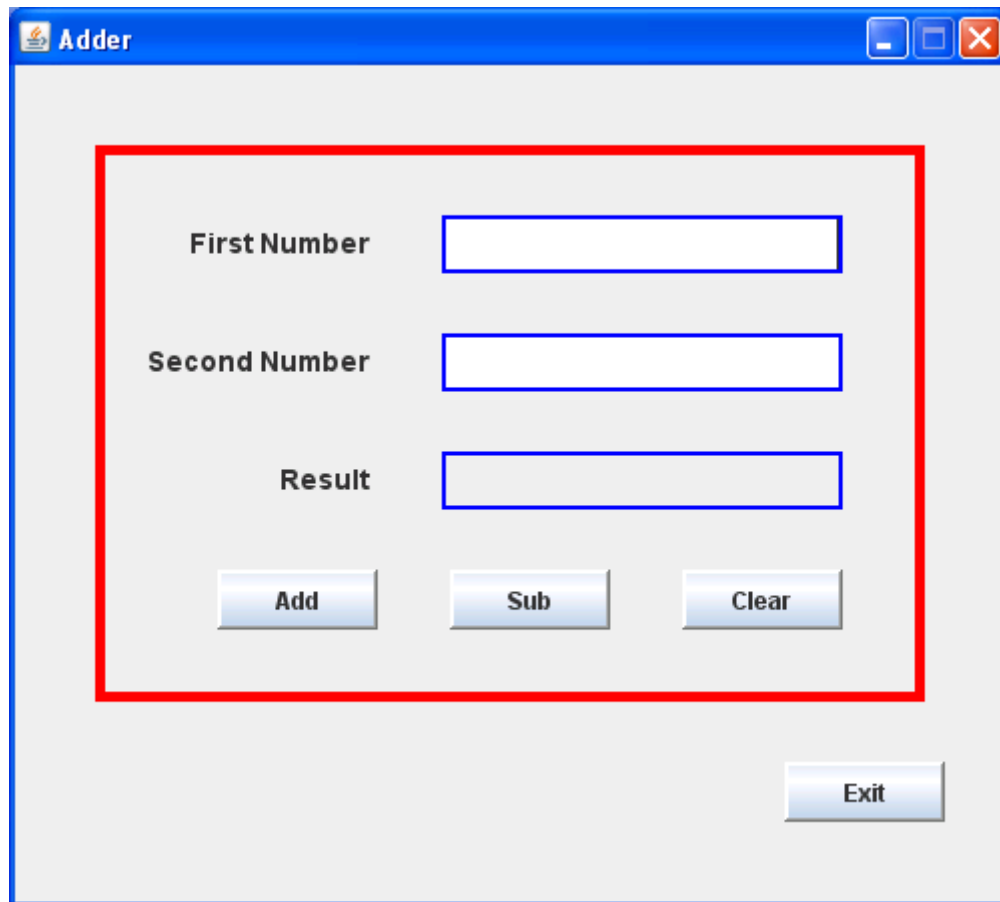
להלן דוגמת שימוש: גודל חלון התוכן הוא (contentWidth × contentHeight) בעוד שגודל התצוגה על המסך היא (windowWidth × windowHeight). אם התצוגה קטנה מן התוכן, יופיעו זחיתים שיאפשרו גלילה:

```
JPanel stuff = new JPanel();
stuff.setSize( contentWidth, contentHeight );
/* ... fill "stuff" with contents no larger than contentWidth X contentHeight */

MyScrollPane msp = new JScrollPane( stuff );

JFrame frame = new JFrame( "MyScrollPane Demo" );
frame.setPreferredSize( new Dimension( windowWidth, windowHeight ) );
frame.add( msp );
/* ... display frame */
```

למשל, ניקח את המחלקה Calculator שבנינו באחד התרגולים: יש בה לא מעט קומפוננטות מסוג JLabel, JTextField, JButton ועוד. היא כולה מוכלת בחלון אחד מסוג JPanel, שאותו אפשר להעביר למופע של JScrollPane שיאפשרו גלילה של התוכן. התמונה להלן מתארת את ה-Calculator כפי שהוא בעצמו:

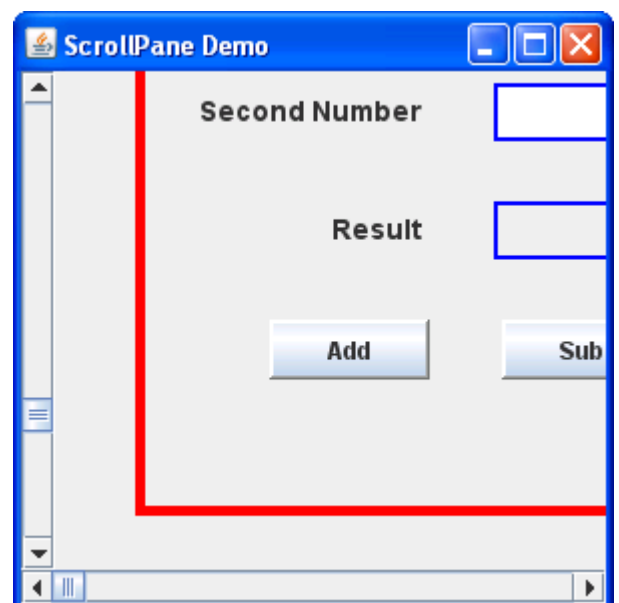


התמונה הבאה מתארת את אותו Calculator כשהוא נתון בתוך JScrollPane קטן יותר. כל הקוד שמייצר את התמונה הקטנה מצוי בצידה.

```
Calculator stuff = new Calculator();
stuff.setSize( 500, 450 );

JScrollPane msp = new JScrollPane( stuff );

JFrame frame = new JFrame( "ScrollPane Demo" );
frame.setPreferredSize( new Dimension( 300, 300 ) );
frame.add( msp );
frame.pack();
frame.setVisible( true );
```



2. עיצוב

כל המחלקות הנזכרות להלן הן מחלקות סטנדרטיות של Swing ורצוי לקרוא את התיעוד שלהן באתר <http://docs.oracle.com/javase/8/docs/api>.

I. ארגון גיאומטרי:

כמו הרבה דוגמאות אחרות שראינו, גם המחלקה JScrollPane יכולה להבנות על JPanel. בתוכו יש להכניס שלוש קומפוננטות:

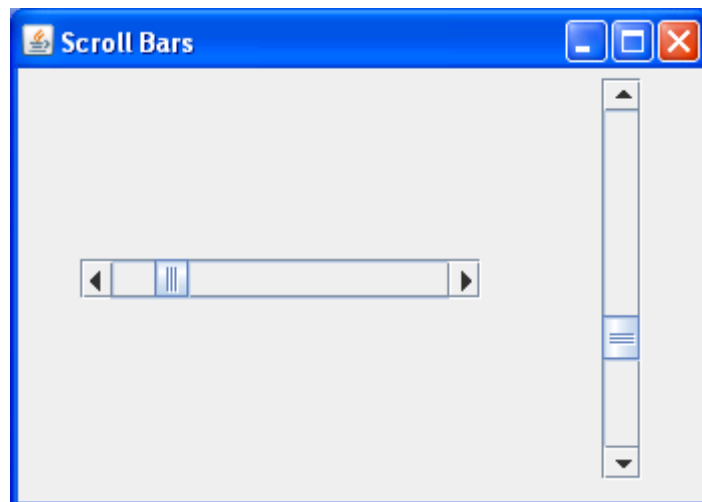
- א. זחית אופקי
- ב. זחית אנכי
- ג. חלון הגלילה

לפי המקובל, הזחית האופקי ישכב בתחתית החלון, הזחית האנכי יעמוד בצידו השמאלי (זהו חלון אנגלי...). ואילו את רוב שטח ה-JPanel יתפוס חלון הגלילה בו מופיע התוכן.

אמרגן (LayoutManager) המתאים לסידור כזה ככפפה הוא ה- BorderLayout שמחלק את החלון לאזור עליון, תחתון, ימני, שמאלי ומרכזי. את הזחית האופקי נשים ב- BorderLayout.SOUTH, את האנכי ב- BorderLayout.WEST ואת חלון התוכן נכניס ל- BorderLayout.CENTER.

II. הזחיתים:

אמנם אין להשתמש במחלקת ה-Swing הסטנדרטית JScrollPane (שמבצעת את כל מה שנדרש מ-JScrollPane), אבל אפשר ורצוי להשתמש במחלקה הסטנדרטית שמייצרת את הזחיתים עצמם – JScrollBar. מחלקה זו יוצרת את הגרפיקה הבאה:



הזחית מימין (האנכי) נוצר כתוצאה מקריאה לבנאי של JScrollBar עם הארגומנט Adjustable.VERTICAL. Adjustable הוא הממשק שמגדיר את הקבועים שבהם מחלקה זו משתמשת. כדי ליצור את הזחית האופקי שמשמאל יש לספק לבנאי במקומו את הארגומנט Adjustable.HORIZONTAL.

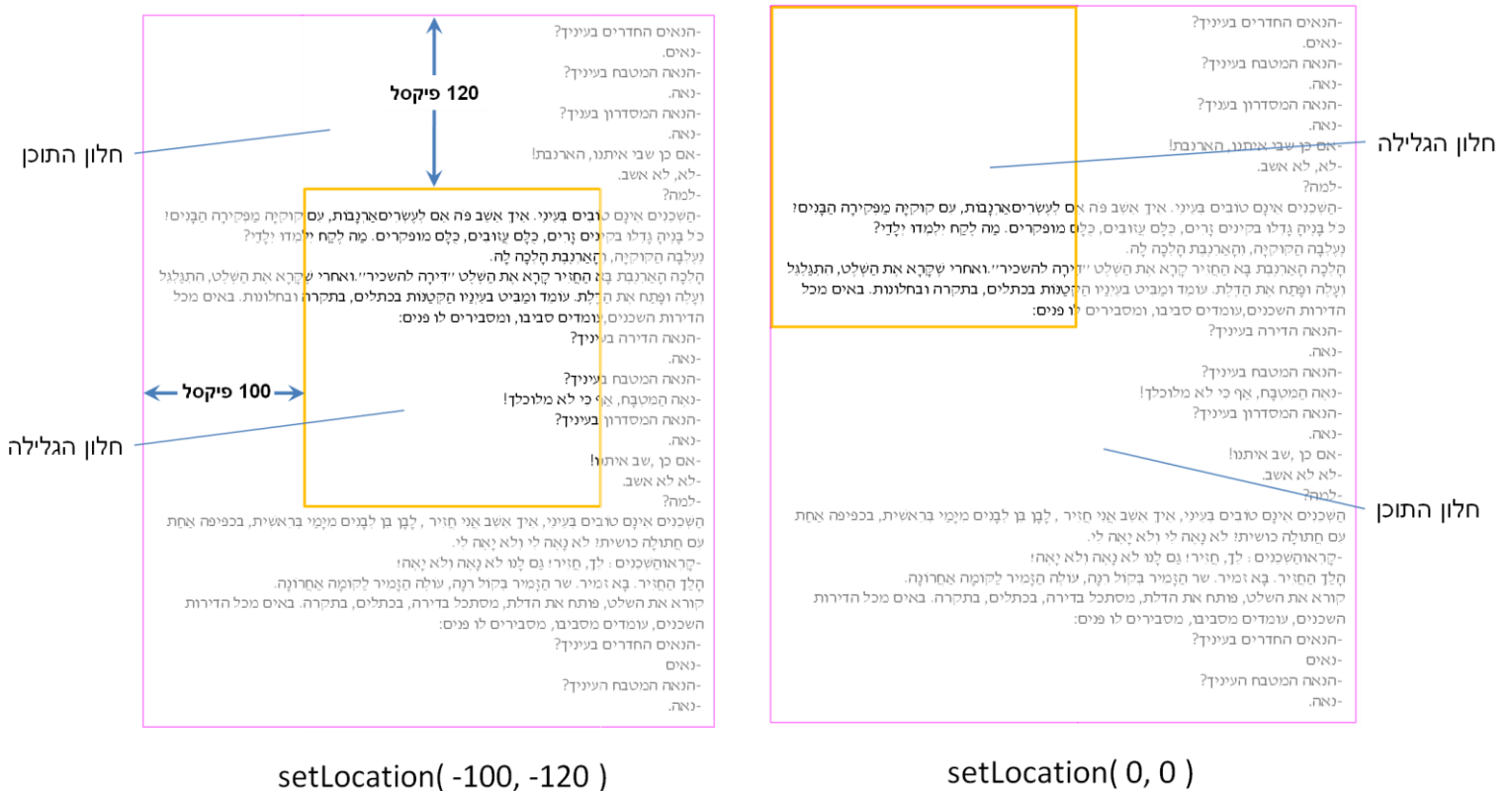
זחחים משמשים באופן כללי לקביעת ערך: כשה"אגודל"¹ של הזחח נמצא לגמרי בצד אחד מתקבל ערך אחד, כשה"אגודל" נמצא בצד האחר מתקבל ערך אחר, ולכל אחד ממצבי הביניים מתאים ערך שנמצא בין שני הערכים האלה, באופן יחסי למקומו של ה"אגודל". את ערכי הקיצון הללו קובעים בהתחלה באמצעות המתודות `setMinimum()` ו-`setMaximum()`. במקרה של `MyScrollPane`, טווח התנועה של הזחח צריך לייצג את טווח התזוזה של התוכן. אפשר לבחור את ערך המינימום כאפס וערך המקסימום כגודל המימד המתאים של חלון התוכן: רוחב החלון עבור הזחח האופקי וגובה החלון עבור הזחח האנכי. אפשר שתעדיפו, אחרי קצת התנסות, לבחור ערכים קצת אחרים.

בנוסף לגרפיקה, בכל פעם שמשתמש מזיז את ה"אגודל", `JScrollBar` גם מייצרת אירוע מסוג `AdjustmentEvent`. כדי לתפוס אותו, יש צורך להגדיר מאזין (`Listener`) מסוג `AdjustmentListener`, ולהעביר אותו לזחח, באמצעות המתודה `addAdjustmentListener()`. הדבר מודיע לזחח על רצוננו להאזין לאירועים מסוג זה.

למאזין חייבת להיות מתודה בשם `adjustmentValueChanged()`. המתודה נקראת בכל פעם שנורה אירוע מסוג `AdjustmentEvent`, והאירוע עצמו מועבר אליה כארגומנט. מהאירוע ניתן להפיק (באמצעות המתודה `getValue()`) את הערך שיש לזחח בעקבות האירוע. מתודה זו היא המקום לגרום לגלילה פרופורציונלית של חלון הגלילה, כפי שנראה בהמשך.

III. חלון הגלילה

חלון התוכן, שהוא על פי רוב גדול, הוא בן של (מוכל בתוך) חלון הגלילה, שהוא על פי רוב קטן יותר (אחרת אין צורך בגלילה כלל). הגלילה נעשית על ידי הזזת הראשית של חלון התוכן יחסית לחלון הגלילה באמצעות המתודה `setLocation()`. ראו ציור להלן:



¹ החלק שבאמצע הזחח שניתן להזיזו (באנגלית `thumb`).

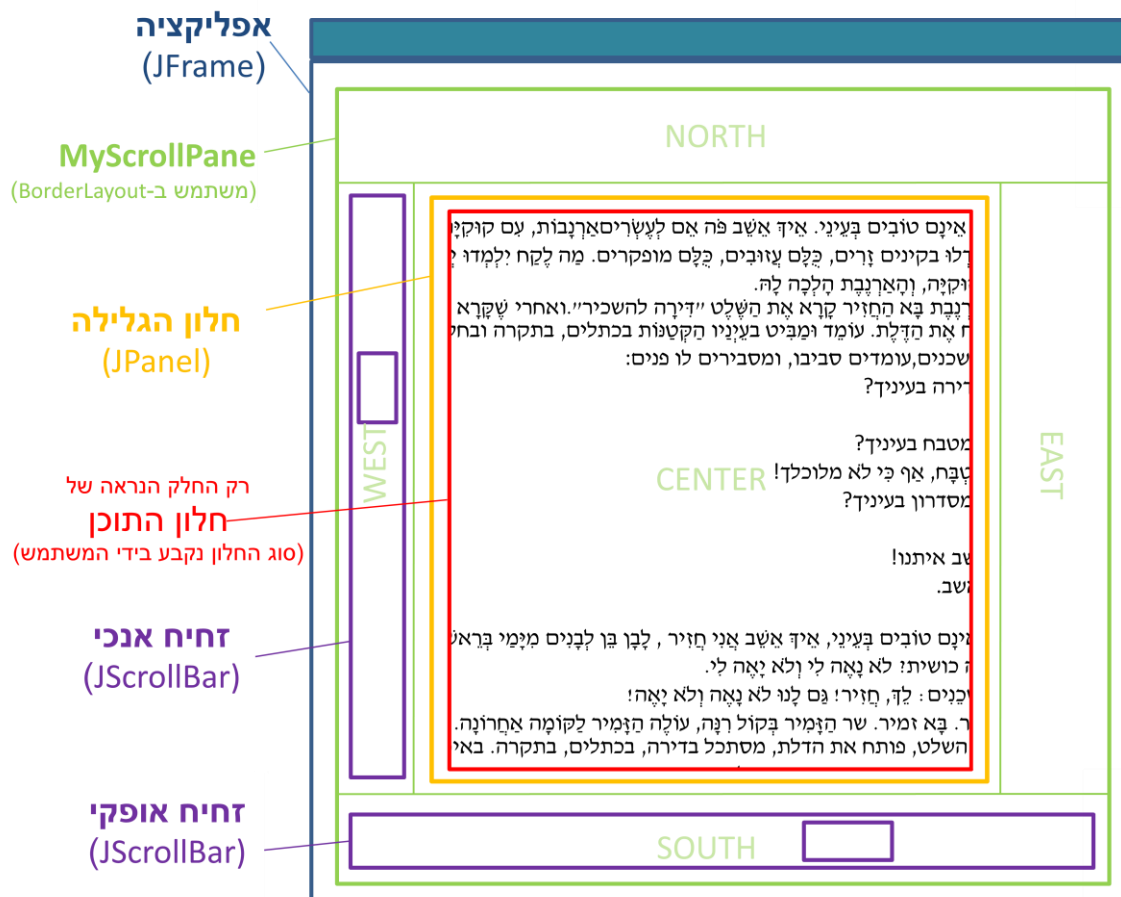
למי שכבר ניסה בודאי גילה ש-Java נוטה להתעלם מהמתודות `setSize()` ו-`setLocation()`. זה קורה משום שבדרך כלל לכל קומפוננטה מוגדר אמרגן (גם אם לא ציינו זאת במפורש – יש ברירת מחדל), ולאמרגן יש עדיפות בקביעת המיקום של קומפוננטות. כדי שהמתודות הנ"ל ישפיעו על התוצאה, יש לבטל במפורש את פעולת האמרגן, באמצעות הגדרת אמרגן ריק, כך:

```
setLayout( null );
```

כדי לשאת את האין-אמרגן הזה נחוץ חלון שיש לנו שליטה עליו. חסרה לנו שליטה כזו בחלון התוכן (המוגש לנו כארגומנט ע"י המשתמש, מוכן וסגור), וגם על אזור האמצע של `MyScrollPane` (שמוגדר ע"י ה-`BorderLayout`) אין לנו שליטה כזו, ועל כן יש להוסיף חלון ביניים.

חלון הגלילה יהיה, אם כן, `JPanel` נוסף שימוקם כבן יחיד במרכז החלון של `MyScrollPane` ויכיל, כבן יחיד, את חלון התוכן. תפקידו יהיה כשל יצחק אבינו: להיות בן לאברהם ואב ליעקב.

הציור שלהלן מדגים את היחסי ההכלה בין הקומפוננטות שתוארו לעיל.



IV. דינמיקה

ישנם שני סוגי שינויים להם נדרש מופע של JScrollPane להגיב:

- א. משתמש הזיז את אחד הזחחים
- ב. משתמש שינה את גודל החלון של JScrollPane

IV-א. הזזת זחח

המקרה הראשון הוא המקרה הרגיל בו משתמש מבקש לגלול את התוכן. הזזת הזחח תגרום לירית אירוע שיגרום מצדו להפעלת המאזין, שבתוכו יש לשתול את הקוד המבצע את הגלילה. יש לשים לב לזחח שהופעל ולשנות רק את שיעור הראשית הקשור לאותו הזחח: אם הזחח שהוזז היה האופקי, יש לשנות רק את שיעור ה-x של ראשית חלון התוכן, כיון שהבקשה היא לגלילה אופקית בלבד. שיעור השנוי הוא הנגדי של ערך הזחח – ככל שערכו גבוה יותר, יש לקבוע את ה-x יותר שמאלה, לערך יותר שלילי. באופן דומה, אם הזחח שהשתנה הוא האנכי, יש לשנות רק את שיעור ה-y. (אפשר תמיד לברר מהם השיעורים הנוכחיים של הראשית באמצעות המתודה getLocation()). אחרי השנוי יש לקרוא ל-repaint() על מנת שהשנוי גם יראה על המסך.

IV-ב. שנוי בגודל החלון

המקרה הזה יותר מורכב: כאן יש שינוי ביחס שבין החלק הנראה של התוכן וגודלו הכולל, ולכן יש לחשב מחדש את המינימום והמקסימום של הזחחים. יתר על כן, יש אפשרות שחלון הגלילה גדל עד כדי כך שכל התוכן כעת גלוי ואין צורך יותר בגלילה, ואפשר לגמרי לסלק את הזחחים. מצב זה אפשרי בכל אחד מהצירים בנפרד: יתכן שרוחב החלון גדול דיו, כך שאפשר לוותר על הזחח האופקי, אבל לא כן האורך, ועל הזחח האנכי להישאר.

שינויים בגודל החלון מדווחים באמצעות אירוע מסוג ResizeEvent, והמאזין השומע אותו הוא מסוג ComponentListener, המחויב להגדיר את המתודה ComponentResized() לשם כך. (יש למאזין זה מתודות נוספות שאותן יש להגדיר, אבל כיון שלא נזדקק להן, נוכל הפעם להגדיר אותן כריקות). צריך, אם כן, לבנות מאזין מסוג ComponentListener ולדווח ל-JScrollPane על קיומו באמצעות המתודה addComponentListener().

נחמה אחת יש בכך שלכל חלון מובטח אירוע מסוג ResizeEvent לפחות פעם אחת, כאשר הוא מוצג על המסך לראשונה. אפשר, אם כן, לקבוע את גבולות הזחחים רק בתוך המתודה ComponentResized() (ולוותר על הגדרתם בהתחלה, למשל, בבנאי). כאמור, מובטח שהזחחים יאותחלו כראוי כיון שאירוע כזה לבטח יירה בטרם יוצג החלון בפעם הראשונה.

במקרה שמסתבר שאחד מממדי חלון הגלילה (או שניהם) גדול מספיק כדי להציג את התוכן (במימד זה לפחות) בשלמותו, ניתן להסיר את הזחח המתאים באמצעות המתודה remove(), שפעולתה הפוכה לפעולת ה-add(): היא מסירה קומפוננטה קיימת.

3. סיכום

זה לא תרגיל פשוט – הוא דורש הבנה של המערכת, ולכן הקדישו לו את הזמן הדרוש: אל תתחילו לחשוב עליו שבוע לפני ההגשה, אלא מיד כשקיבלתם אותו. בתמורה, אחרי ההגשה תדעו על המערכת החלונאית כל מה שרציתם לדעת ולא העזתם לשאול...

הערה חשובה: לא לכל חלון יש ממדים, גם אם החלון מוצג על המסך (וממדיו בעליל אינם אפס). אם שלחתם חלון להצגה על המסך אך הוא אינו מופיע, בררו את ממדיו (באמצעות הדפסת ביניים של getWidth() ו-getHeight()) וודאו שאינם אפס. הקפידו לתת ממדים לחלון התוכן באמצעות setSize().

נ.ב. אחד הסטודנטים הציע לבצע את הגלילה ע"י גרירת חלון התוכן עם העכבר : הקלקה בנקודה כלשהי בחלק הנראה של התוכן וגרירתו (ללא שחרור לחצן העכבר) למקום אחר. גרירה שכזו מוגבלת אמנם לתחומי חלון הגלילה, ומחייבת כמה גרירות אם חלון התוכן גדול דיו, אבל מייטרת את השימוש בזחיחים. צורת אינטראקציה זו היא אולי יותר אינטואיטיבית למשתמש, אך דורשת טיפול באירועים מסוגים אחרים מאלה שתוארו לעיל. עם זאת, תכנית שתממש צורת אינטראקציה זו בצורה נכונה תתקבל כפתרון לתרגיל.

בהצלחה!