

Known Issues

ראשית לשם הבנה כללית של התכנית אצרף את מטלת הפרוייקט מקורס 10118 (עשיתי את הקורס בסמסטר א' הקודם אצל ויקטור טאובקין). מדובר בהעמסת קונטיינרים על גבי אונייה באמצעות כפתור move. ניתן להעמיס מס' מכולות אחת על גבי השנייה. כפתור restart יאתחל את האונייה שבנמל מחדש.

הערות כלליות לגבי השינויים שנעשו בפרוייקט המקורי לטובת המטלה:

- בpackage "application" במחלקה "Controller" שיניתי את setOnAction כך שיתפקד על ידי lambda expression במקום בדרך הארוכה. כמו כן במחלקה זו מופיעים האתחולים עבור כל אחד מהכפתורים מהחלונית עם הGUI. היצירה עצמה של הכפתורים נמצאת בpackage "View" במחלקה "MainView" כנהוג במודל MVC.
- בpackage "Model" במחלקה "Port" בשורה 138 - במקור, ההגרלה הרנדומלית עבור הייצוג העשרוני של כל גוון צבע למכולה (מודל RGB) היה בטווח הערכים המקסימאלי (0 עד 256), אך לטובת סעיף 13 בו נדרשת כפילות בין שתי תכונות במספר מופעים של האובייקט, ביצעתי שינוי בערכי redn והgreen של האובייקט כך שיוגרלו בין 0 ל 2, מה שגרם למכולות להיצבע כעת בגווי כחול במקום בצבעוני. (אם היה נשאר כמקור כך שעבור כל ערך יוגרל טווח הצבעים המקסימאלי - היה סיכוי קלוש ששני גוונים ייצאו זהים לשני אובייקטים בשתי תכונות כמבוקש, ובשאר התכונות שאינן צבע לא ייתכן שיוויון בין שני אובייקטים).
- בpackage "Model" במחלקה "Container" בשורות 142 ו151 הוספתי מתודות hashCode ו-equals עבור סעיף 13. השיטות הללו לא היו קיימות ולא היה בהן צורך בפרוייקט המקורי.
- כל שאר המתודות שנוספו עבור המטלה נמצאות בpackage "Model" במחלקה "Model" החל משורה 136.

כעת אתייחס לכל אחד מהסעיפים במטלה הנוכחית:

12. עבור ביצוע הסעיף הוספתי בחלונית של JavaFX כפתור בשם "add to collection 1". בלחיצה על הכפתור, מופעלת המתודה copyContainersToColl1() (נמצאת בpackage "Model" במחלקה "Model" בשורה 138) והקונטיינרים שעל מחסנית הנמל מועתקים לתוך TreeSet. בנוסף מודפסים לconsole כל פרטי הקונטיינרים שנמצאים כעת על האונייה בסדר יורד לפי תכונת left שלהם (מעין ערך הx של הפינה השמאלית העליונה של המכולה). נשים לב שאם הועברו כבר חלק מהמכולות אל המשאית ולוחצים שוב על הכפתור "add to"

1 collection" אז מודפסים לconsole רק הקונטיינרים שכעת על האונייה, פחות אלו שכבר על המשאית או לצד הדרך. המיון מבוצע בעזרת comparator.

13. עבור ביצוע הסעיף הוספתי בחלונית של JavaFX כפתור בשם "add to collection 2". בלחיצה על הכפתור, מופעלת המתודה copyContainersToColl2() (נמצאת בpackage "Model" במחלקה "Model" בשורה 145) והקונטיינרים שעל מחסנית הנמל מועתקים לתוך TreeSet וממנו לתוך HashSet. בלחיצה זו נוספים לHashSet ומודפסים לconsole רק פרטי הקונטיינרים כך שאם קיימת מכולה/מכולות שערכי תכונות red והgreen שלהם/ם שווים לאותם ערכי red והgreen של מכולה/מכולות אחרים, ייכנס ויודפס רק אחד מהם, כלומר HashSet לא יכניס איברים חדשים שהם כפילות עם מה שכבר נמצא בו. הברירה נעשית בעזרת hashCode ו- equals.

14. כל אחת מהשיטות לעיל מפעילה את המתודה print(Set<?> set) אשר מדפיסה את תוכן ההcollectionn שהתקבל כפרמטר בעזרת iterator.

15. כפי שציינתי מעלה, בpackage "application" במחלקה "Controller" שיניתי את ההsetOnAction כך שיתפקד על ידי lambda expression במקום בדרך הארוכה.

נספח: הפרוייקט מקורס 10118:



פרויקט סיום בקורס "תכנות מונחה עצמים".

הנחיות טכניות להגשה

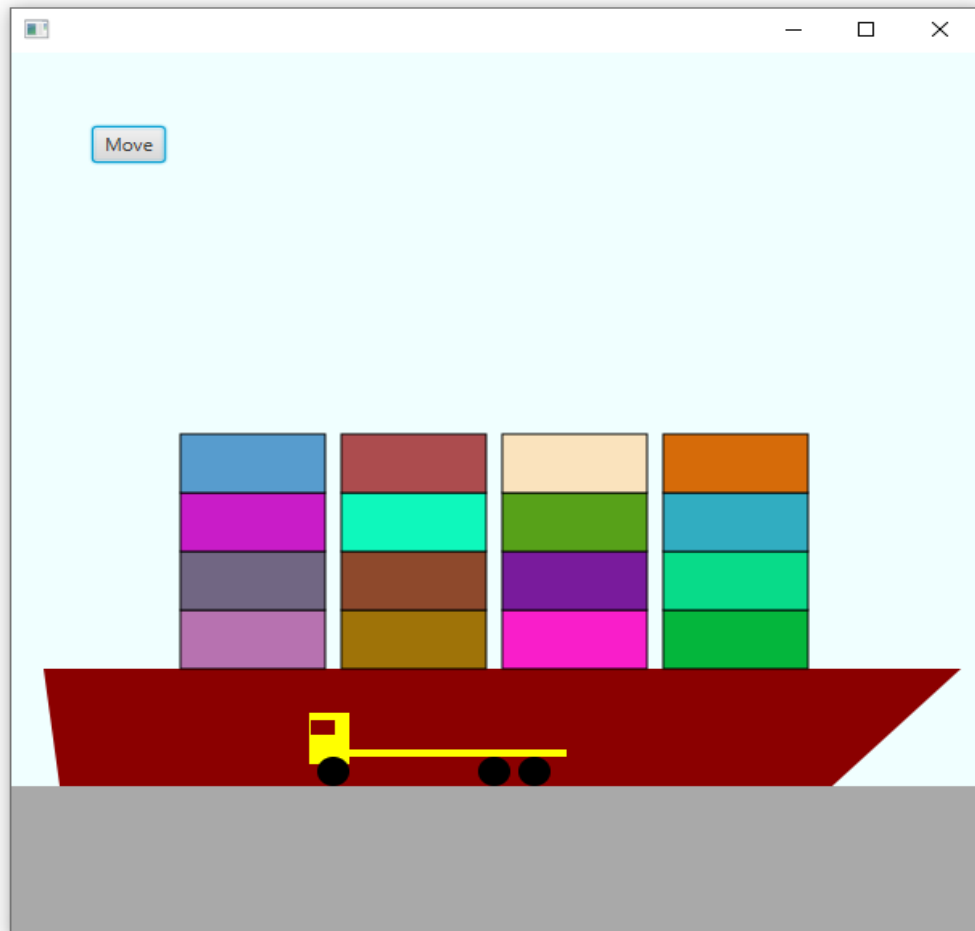
1. הגשת הפרוייקט בתיבת ההגשה במודל עד התאריך המצוין בתיבה.
2. ניתן להגיש את הפרוייקט בזוגות או ביחיד.
3. רק אחד מבני הזוג מעלה את הפרוייקט למודל

מה עליכם לעשות?

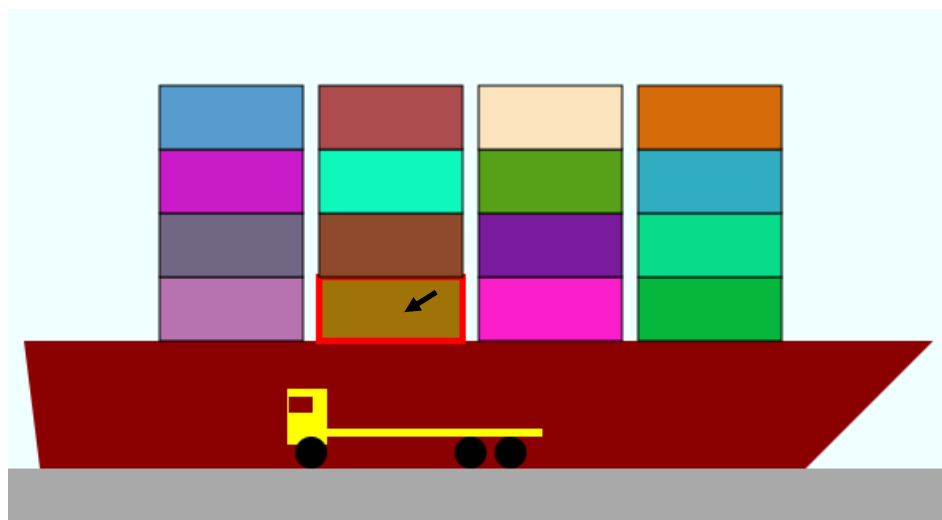
כתבו תכנית המשתמשת ב-javafx ובתבנית MVC לפתרון הבעיה המתאורת בהמשך. יש להשתמש בעקרונות של תכנות מונחה עצמים כפי שנלמד בכיתה.

תיאור הבעיה

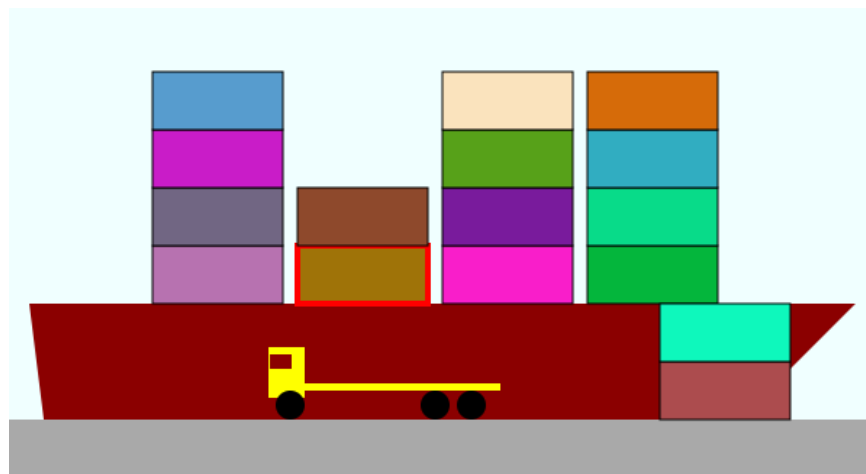
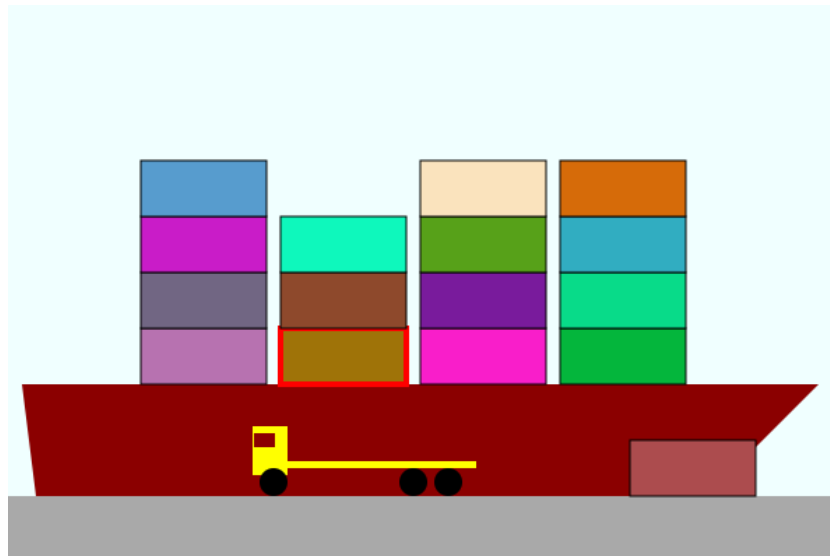
בנמל עוגנת ספינת מכולות ועל המזח נמצאת משאית אשר יכולה להעביר מכולה אחת בלבד למגרש לאחסון מכולות.

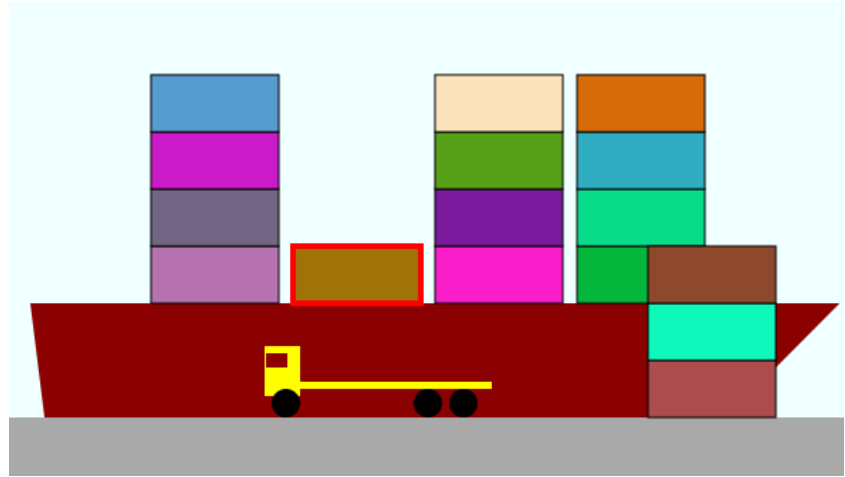


משתמש המערכת צריך לסמן מכולה (לאו דווקא עליונה) בעזרת קליק העכבר ולאחר מכן להתחיל ללחוץ על לחצן "Move" כדי לראות איך המערכת מוציאה את המכולה מהספינה למשאית. המנוף (שאין צורך לציירו) יכול להעביר בפעולה אחת מכולה אחת מסיפונה של הספינה אל המזח או על המשאית. ניתן לשים מכולה על מכולה אחרת. להלן סימון המכולה:

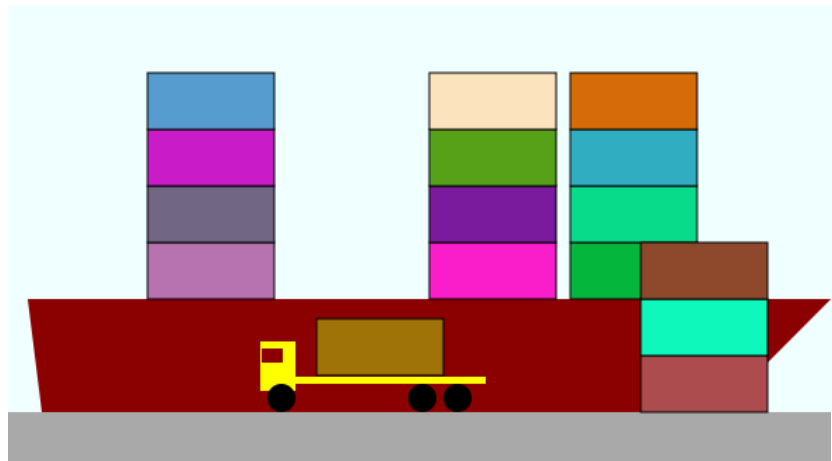


תמונות הבאות מציגות תוצאות של לחיצות עוקבות על לחצן "Move". שימו לב, כי המכולות נאספות על המזח בסדר הפוך מזה שהיה להן על הסיפון.

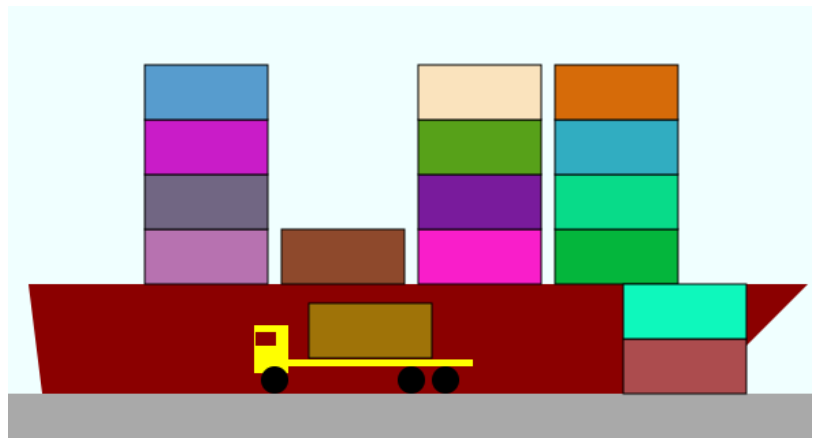


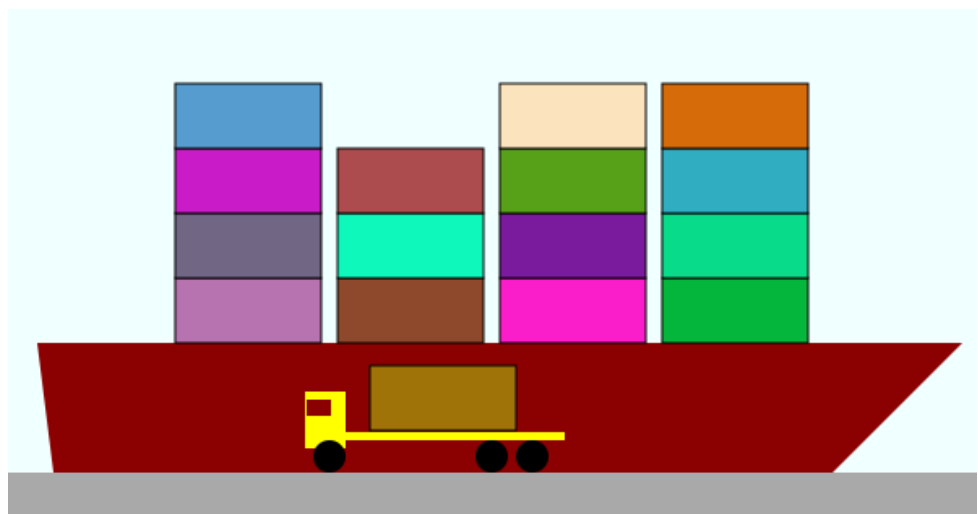
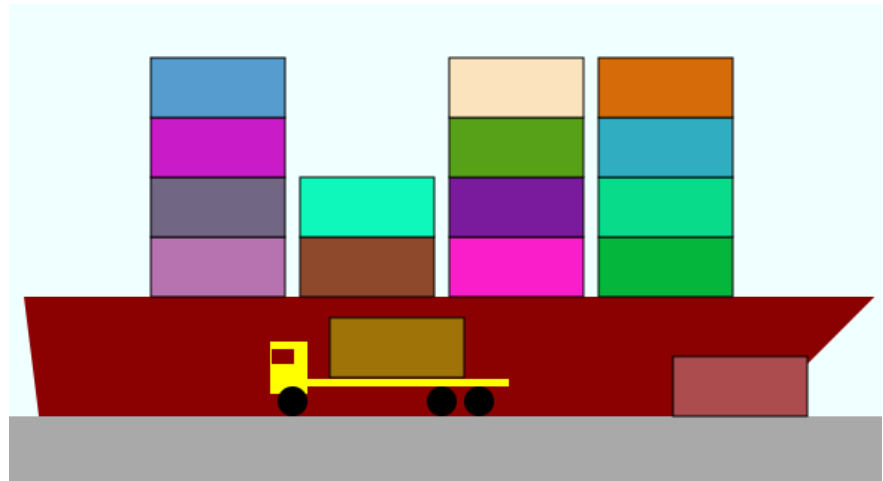


כאשר מגיע תורה של המכולה המסומנת על ידי המשתמש היא עוברת ישירות אל המשאית:



לחיצות נוספות על לחצן "Move" מחזירות את המכולות מהמזח חזרה אל הסיפון





בהצלחה!

נספח 2: המטלה הנוכחית בקורס 10119:

חלק 1: Collections, Generics, Iterator, Lambda Expression

12. יש להוסיף אפשרות בתפריט, שהפעלתה תעביר את אחד מהמערכים של אוביייקטים / ArrayList הקיימים, לבחירתך, ל- Collection הכי מתאים, כך שבכל שלב אם נרצה להדפיס את תוכן ה- Collection הזה, נקבל תמיד סדר לפי מיון יורד, כאשר המיון נקבע לפי אחת התכונות באובייקט לבחירתך.
- יש לבחור תכונה שהיא לא מספר מזהה או משהו ייחודי לכל אובייקט, אלא תכונה עם ערך שחוזר על עצמו פה ושם באובייקטים השונים.**
- יש לבצע כך, שכל האיברים מהמערך המקורי יוכנסו תמיד, אפילו אם כל התכונות שלהם זהות לחלוטין. יש להדפיס את תוכן ה- Collection בעזרת Iterator כפי שלמדנו בכיתה.
13. כעת, יש להוסיף אפשרות בתפריט, שהפעלתה תעביר את כל תוכן ה- Collection הקודם ל- Collection נוסף מתאים, כאשר הדגש יהיה כעת על המהירות (הוספה / חיפוש / הסרה), ואין משמעות למיון כלשהו. בנוסף, ה- Collection הפעם לא יכניס איברים חדשים שהם כפילות עם מה שכבר נמצא בו. כפילות מוגדרת לפי שוויון מוחלט בין 2 תכונות לבחירתך.
- יש לבחור תכונות שהן לא מספר מזהה או צירוף ייחודי לכל אובייקט, אלא תכונות עם צמד ערכים שחוזרים על עצמם פה ושם באובייקטים השונים.**
14. יש להדפיס את תוכן ה- Collection בעזרת Iterator, כפי שלמדנו בכיתה, ורוצים שההדפסה תהיה לפי סדר ההכנסה של האיברים ל- Collection החדש, כלומר לפי הסדר המקורי שהיה בעצם עם המיון, אבל בלי הכפילויות.
15. בקוד שאתה מוסיף, באם יש צורך, וניתן, אזי יש לעבוד עם Lambda Expression.
16. בקובץ ה- Known Issues, יש להוסיף הסבר, באיזה מערך / ArrayList מקורי השתמשתם מתוך הפרויקט, איפה הוא נמצא בקוד (שם הקובץ, מספרי שורות וכו'), ולפי איזה תכונה מתבצע המיון, ולפי איזה תכונות נקבעת הכפילות ב- Collection השני, וכיצד להפעיל את שני הסעיפים הנ"ל דרך התפריט או דרך איזה כפתורים.