מטלה 4

הנחיות כלליות:

- מטלה זו נעשה ביחידים בלבד. אסר למסור או לקבל כל קוד מגורם אחר, ואפילו שורה אחת.

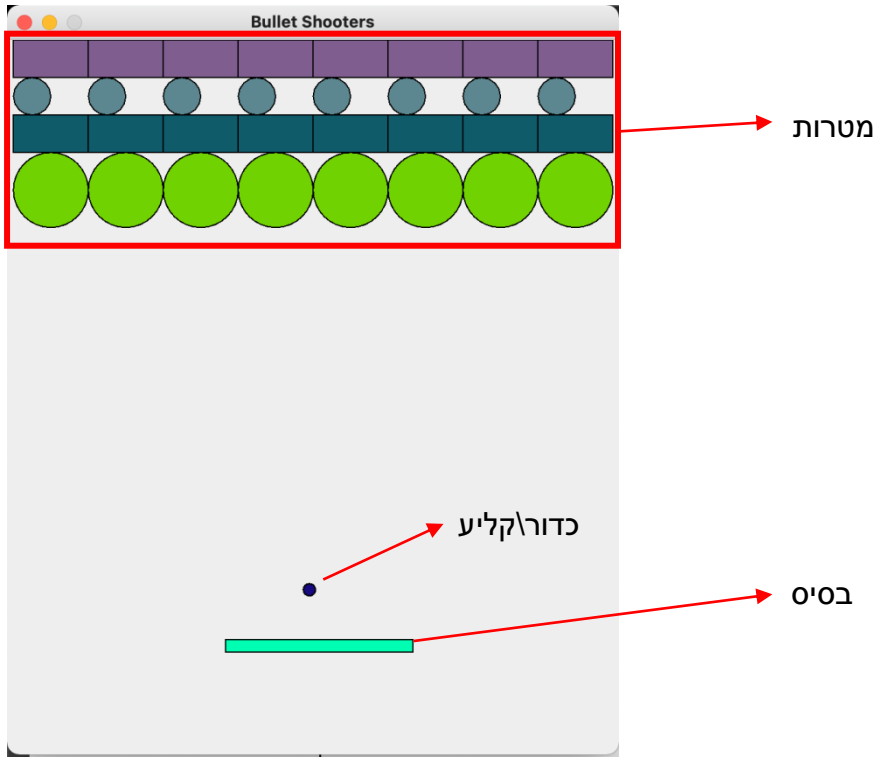
- במטלה אתם מתבקשים למלא ארבעה קבצים, Node.java, LinkedList.java, Rectangle.java, Circle.java. יש להקפיד ששמות הקבצים הם בדיוק כמבוקש. יש לדחוס את הקבצים האלה לקובץ zip ששמו הוא מספר הזהות של התלמיד בלבד. יש להקפיד ששם הקובץ יהיה בדיוק כמבוקש. אין לצרף קבצים אחרים להגשה.

הקדמה:

במטלה זו ניצור בסיס של משחק שרוב מכם בוודאי כבר מכירים. נקרא לו Bullet Shooter. המשחק מורכב משלושה חלקים בסיסיים: בסיס (base), קליע\כדור (bullet), ומטרות (targets). נפרט כל אחד מהמרכיבים בהמשך. להלן הנחיות המשחק:

- כאשר הכדור "פוגע" באחת המטרות הוא משמיד אותה, ומשנה כיוון (חוזר אחורה).
- כאשר הכדור "נוגע" חזרה בבסיס הוא "נורה" בחזרה לעבר המטרות.
- השחקן יכול להזיז את הבסיס לכל הכיוונים עם תנועת העכבר.
- השחקן מנצח כאשר הוא משמיד את כל המטרות.
- המשחק מתחיל בעת לחיצת ENTER במקלדת.

חזית המשחק:



למטלה זו מצורפות שתי המחלקות הבאות, שאותם **<u>אין</u>** לשנות:

- GameBox.java – מנהלת את בניית חזית המשחק ומחזיקה את כל האירועים (הזזת עכבר, לחיצה על כפתור ENTER).
- RunGame.java – אותה מריצים כדי להתחיל את המשחק.

כמו כן, מצורפים שלשת הממשקים, שאותם **<u>אין</u>** לשנות:

- GeoShape.java – ממשק המייצג צורה גאומטרית, הכולל את השיטות הבאות:

1. int getX(), int getY() - מחזירות את ערכי x,y של הצורה.
2. void setX(int x), void setY(int x) – מקבלות כפרמטרים ערכי x ו- y בהתאמה ומשנות את השדות המקבילות של הצורה בהתאם.
3. void setDx(int dx), void setDy(int dy) - מקבלות ערך הזזה ל-x או y ושומרת אותם.
4. int getDx(), int getDY() – מחזירות את ערך ההזזה האחרון של x או y.
5. void translateX(), void translateY() - משנות את הערך של x או y , כלומר מוסיפות את ערך הזזה האחרון לערך של-x או y
6. void setColor(int r, int g, int b) - מקבלת כפרמטרים שלושה ערכים של אדום, ירוק וכחול, ומשנה את השדות המקבילות של הצורה בהתאם.
7. int getRed(), int getGreen(), int getBlue() – מחזירות את הערכים של אדום, ירוק או כחול.
8. boolean intersects(GeoShape g) - מקבלת כפרמטר צורה אחרת ומחזירה true אם הצורה המתקבלת חותכת (חופפת) את הצורה הנוכחית, ו- false אחרת. תשימו לב שאתם חייבים להבדיל בין סוגי הצורות השונות.
9. void draw(Graphics g, Component c) – בגדול, מציירת את הצורה על המסך. כבר מימשנו לכם את הפונקציה הזאת, ואין לשנות אותה!

- NodeInterface.java – ממשק המייצג חוליה ברשימה מקושרת, הכולל את השיטות הבאות:

1. GeoShape getData() – מחזירה את הצורה ששמור בחוליה. על כל חוליה להכיל צורה גיאומטרית כמידע שמור.
2. setData(GeoShape g) – מקבלת כפרמטר צורה גיאומטרית, ושומרת אותה בתור המידע של החוליה.
3. NodeInterface getNext() – מחזירה מצביע לחוליה הבאה.
4. void setNext(NodeInterface next) - מקבלת כפרמטר מצביע לחוליה הבאה ושומרת אותו בתור החוליה הבאה.

- LinkedListInterface.java – ממשק המייצג רשימה מקושרת, הכולל את השיטות הבאות:

1. NodeInterface getHead() – מחזירה מצביע לראש הרשימה, כאשר כל חוליה ברשימה הינה מסוג NodeInterface.
2. void remove(NodeInterface p) – מקבלת כפרמטר מצביע על חוליה ברשימה, ומוחקת אותה מהרשימה.

המטלה:

1. **השלימו את המחלקות:**

- Circle.java - המייצגת מעגל במישור, כאשר לכל מעגל נחזיק: x, y – שני מספרים שלמים שמייצגים את הקואורדינטות של נקודת מרכז המעגל, dx,dy – ערכי הזזה ל-x,y, radius – מספר שלם שמייצג את הרדיוס של המעגל, ומספרים שלמים r,g,b המיצגים את צבעי התמונה. המחלקה תיישם את הממשק Geoshape.
- Rectangle.java – המייצגת מלבן במישור שצלעותיו מקבילות לצירים, כאשר לכל מלבן נחזיק שבעה מספרים שלמים: x, y, dx, dy, width, height, r,g,b, כאשר x, y מייצגים את הקואורדינטות של הפינה השמאלית העליונה של המלבן, dx,dy הם ערכי הזזה ל-x,y, width, height מייצגים את רוחב ואורך המלבן בהתאמה. r,g,b מיצגים את צבעי התמונה. המחלקה תיישם את הממשק Geoshape.
- Node.java – מחלקה המייצגת חוליה ברשימה מקושרת המממשת את הממשק NodeInterface.java.
- LinkedList.java – מחלקה המייצגת רשימה מקושרת המיישמת את הממשק LinkedListInterface.java – הרשימה אמורה להחזיק לנו את כל המטרות.

תריצו את המשחק ותוודאו שהכל עובד כמו שצריך, ורק אז תגישו.

ב ה צ ל ח ה !