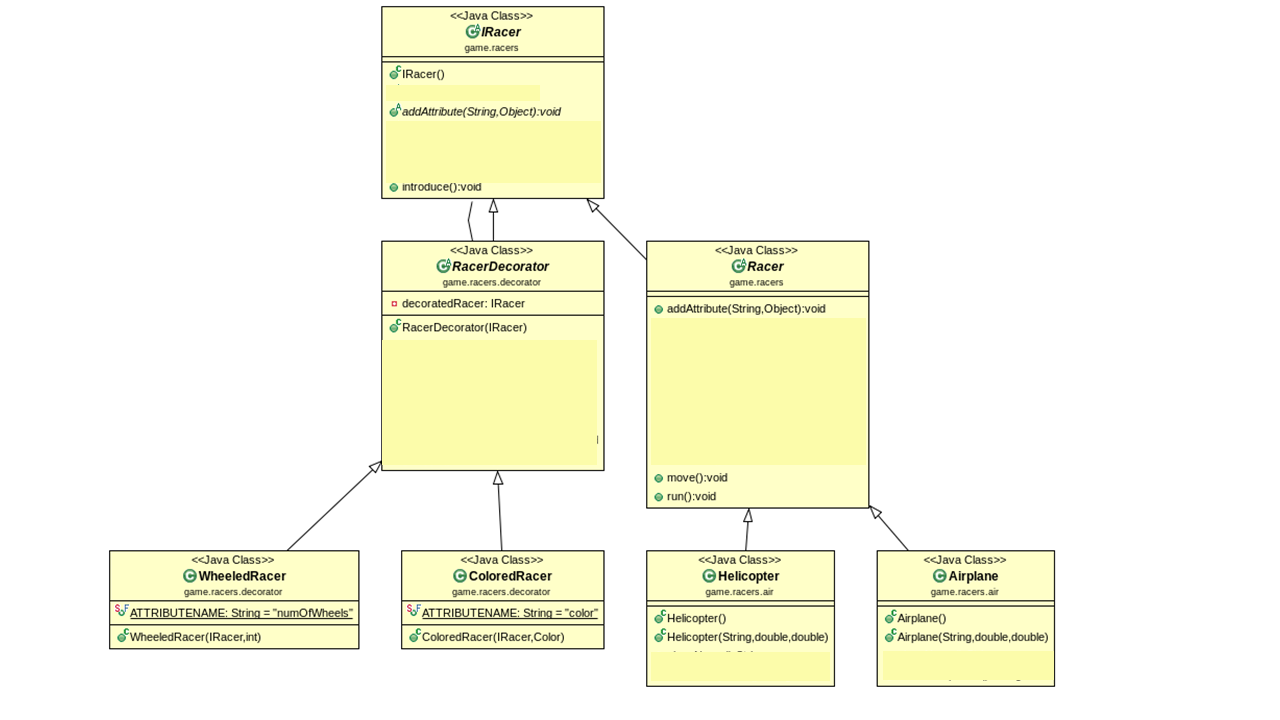
**תכנות מונחה עצמים מתקדם**

**עבודת הגשה מס 4'**

**להגשה עד ה- 7/06/2018 ב- 23:55**

**נא לקרוא את כל המסמך עד סופו לפני תחילת העבודה!**

**דגשים לעבודה זו**

* הערה חשובה: לבחירתכם, ניתן לוותר על ה-GUI בעבודה הזאת ולהחליף אותו באינטראקציה עם STDIN ו-STDOUT (כלומר, לקבל קלט מהמשתמש דרך ארגומנטים ולהראות כל התקדמות/אירוע ע"י הדפסת פלט על הקונסול. מי שיחליט לעבוד עם GUI יקבל **בונוס עד 20 נקודות** לציון העבודה (ולכן יש דרישה לעבוד בצורה עצמאית ללא עזרה מצוות הוראה)
* העבודה מתבססת על עבודת הגשה 3 – עליכם לעדכן את הקבצים הקיימים ולהשתמש במחלקות שכבר בניתם. יש להעלים את המחלקות/ממשקים שהתבטלו בעקבות השינויים.
* עליכם להוסיף מימוש של כמה תבניות עיצוב, כגון:
  + **Decorator** – בשלב של הוספת מתחרה חדש, יש לאפשר למשתמש לבנות מתחרה "מותאם אישית" ע"י הוספת גלגלים וצבע. ההוספה הזאת יכולה להתבצע על כל סוג של מתחרה, בכמות וסדר חופשיים (למשל, להוסיף גלגלים פעמיים לסירה ולצבע באדום).
    - יש להוסיף מחלקות לפי דיאגרמת UML (הדיאגרמה אינה מלאה והרבה מתודות ושדות מוסתרים):
    - 
    - כל עיצב (decoration) חדש מתבצע ע"י עדכון של טבלת המאפיינים (מומלץ להשתמש בטיפוס מובנה HashTable) שמוגדרת בכל מתחרה ומאותחלת בבנאי שלו עם המאפיינים ה"דאפולטיביים".
    - יש לוודא שמימשתם את התבנית בצורה הנכונה ע"י הרצת קוד המצורף לעבודה.
    - העיצוב של מתחרה אמור להשפיע על התהליך של בניית (הרכבת) מרוץ בשלב של הוספת מתחרה לזירה, לפי החוקים החדש:
      * ל-LandArena ניתן להוסיף כל מה שמוגדר תחת racers.land וגם כל מתחרה בעל גלגלים
  + **Prototype** – יש לאפשר למשתמש דרך קיצור בהוספת מתחרה - להעתיק מתחרה הקיים ולשנות תכונות (צבע ומספר). להזכרתכם, מספר אמור להיות ייחודי לכל מתחרה.
  + **Factory Method** – לממש פקטורי עבור יצירת זירות. פקטורי יקבל פרמטר (Land, Navy, Air) שיקבע טיפוס הזירה הנוצרת. הערה: ניתן לטעון דינאמית שמות החבילות ולהציע למשתמש לבחירתו.
  + **Builder –** יש להשתמש בתבנית הזאת ע"מ להרכיב מרוץ. כל מרוץ מורכב מזירה ורשימת מתחרים. יש לספק בילדר דפולטיבי ע"מ להרכיב מרוץ מכוניות (CarRaceBuilder). הבילדר הדאפולטיבי אמור לקבל כמות המתחרים N כארגומנט ולבנות מרוץ מזירה מסוג LAND ו-N רכבים עם הגדרות דאפולטיביות. יש להשתמש ב-Prototype ע"מ "לשכפל" רכבים ופקטורי עבור יצירת זירה.
  + **Observer –** יש לשנות את המימוש הקיים (רק לאלו שלא ביצעו את זה בעבודה 3 ומעוניין לקבל בונוס בעבודה הזאת!) כך שממשק גרפי (GUI) יהפוך ל-Observer וכל מתחרה יהפוך ל-Observable שלו. עדכון המיקום של מתחרה אמור להתבצע ע"י "הודעה" אוטומטית ל-Observer כל מ"מ של תזוזה (או מרחק משמעותי אחר – לשיקולכם).
    - הערה: שימו לב, שאחרת, אם נעשה עדכון לפי זמן (כמו שזה הוגדר בעבודה 3, לא יהיה צורך ב-Observer ומימוש יהיה הרבה פחות יעיל. כמו כן, תנסו להימנע מעדכון של כל החלון (פעולה יקרה מדי) אלא תבצעו repaint() רק עבור חלק הרלוונטי בממשק.
  + **State** – יש להבדיל בין 4 מצבים שונים של מתחרים: פעיל (active), שבור (broken), פסול (disabled) והושלם (completed). בכל שינוי המצב, יש להודיע לזירה (היא ה-Observer). זירה אמורה "להגיב" בהתאם למצב החדש של המתחרה באופן הבא:
    - במקרה של מצב **פעיל** זירה מציגה (דרך console או GUI) את הדירוג הנוכחי של המתחרה (המקום היחסי שלו בין כל המתחרים)
    - במקרה של מצב **שבור** זירה מציגה (דרך console או GUI) את הודעה על כך שהמתחרה שבור וזמן שבו הוא נשבר (יחסי להתחלת המרוץ)
    - במקרה של מצב **פסול** זירה מציגה (דרך console או GUI) את הודעה על כך שהמתחרה פסול ומעדכנת את טבלת התוצאות עם ערך Failed בשורה שלו
    - במקרה של מצב **הושלם** זירה מציגה (דרך console או GUI) את הודעה על כך שהמתחרה סיים את המרוץ ומעדכנת את טבלת התוצאות עם התוצאה שלו
    - הערה: מומלץ להפיל THREAD DAEMON ע"מ לעדכן טבלת התוצאות ולמיין את שורות של מתחרים בה לפי תוצאות הנוכחיות (מיקום יחסי לפני שמתחרה סיים ומקום יחד עם התוצאה אחרי שהוא סיים).

**עבודה נעימה!!!**