מבוא לתכנות מונחה עצמים
עבודת הגשה מס ' 1

מבוא

זהו התרגיל הראשון בקורס ומהווה בסיס לסדרת תרגילים שיינתנו בהמשך הקורס, בנושא בקרת תנועה.

מטרת התרגיל היא כתיבת מחלקות המייצגות דרכים, צמתים וכלי רכב.

נא לקרוא את כל המסמך לפני תחילת העבודה!

<u>דגשים להגשה</u>

- ניתן להגיש עבודה זו בזוגות – רק אחד מהסטודנטים יגיש את העבודה באתר. בכל קובץ יש לציין שם ומס' ת.ז. של מגיש\ים, בתוך תיעוד ה javadoc.
- יש להגיש את העבודה כתיקייה מכווצת בפורמט zip. כאשר שם הקובץ המוגש מורכב ממספרי ת.ז. של המגישים, מופרדים ע"י קו תחתון. לדוגמה: 123456789_987654321.zip
- חלק מניקוד העבודה מתבצע ע"י בדיקות אוטומטיות ולכן חשוב מאוד להגדיר את כל המחלקות, המתודות והשדות בדיוק כפי שצוינו במסמך.
- על הקבצים המוגשים יש לעבוד בצורה מדויקת עם קובץ Program אותו תקבלו יחד עם המשימה תוך כדי התאמה מלאה לדוגמאות ההדפסה המסופקות.
- לכל שאלה לגבי העבודה ניתן לפנות למתרגלת במייל krsofi@gmail.com

<u>דגשים לעבודה זו</u>

- עבודה זו מהווה בסיס לשאר העבודות בקורס, בעבודות הבאות תתבקשו לבצע תוספות ושינויים לעבודה זאת.
- <u>על כל השדות בכל המחלקות להיות פרטיים בלבד. גישה אליהם תתבצע בעזרת get/set .</u>
- <u>לכתוב בכל מחלקה את הבנאים הדרושים כך שתעבוד המחלקה הראשית Program .</u>
- על כל הקבועים להיות תחת שדות final. שימו לב, שכל ההתייחסויות במהלך הקוד לערכים ספציפיים ייעשו באמצעות גישה לקבועים הנ"ל.
- ניתן ליצור משתנים ומתודות במחלקות לשיקולכם, אבל אין לבצע שום שינוי בקובץ Program.
- דוגמאות הרצה הן <u>מחייבות</u> והפתרון שלכם צריך להפיק את אותו הפלט בדיוק פרט למקומות בהם מדובר על ערכים רנדומליים.

<u>תיאור המערכת:</u>

הפרויקט עוסק במערכת בקרת מוניות. בתחילת ההרצה נוצרת מפה, המורכבת מצמתים וכבישים שמחברים ביניהם. כמו כן, נוצרים רכבים וכל רכב מקבל נקודות מוצא ומסלול רנדומלי (מונית פנויה שנוסעת ברחבי העיר). בשלבים הבאים של הפרויקט יתווספו אלמנטים חדשים למבנה זה, כגון לקוחות שמזמינים מונית לטובת נסיעה מנקודה מסוימת לנקודה אחרת, חלוקת הזמנות בין המוניות ועוד...

<u>עליכם לממש את המחלקות הבאות:</u>

התוכנית מתנהלת לפי שעון מדומה – לולאה שרצה ומקדמת את הערך של "זמן", בכל פעימת "שעון" אובייקטים של מחלקות שונות מבצעים את הפעולות הנדרשות (רמזורים מתחלפים, רכבים מבצעים נסיעה למרחק כלשהו בהתאם למהירות שלהם וכדו').

1. מחלקת Point

- המחלקה מייצגת נקודה על מערכת צירים דו מימדית. ציר X נע בין הערכים 0 ל-800, ציר Y נע בין 0 ל-600.
- למחלקה מתודה הבודקת את תקינות ערכי X, Y וכאשר מתבצע ניסיון ליצור מופע של מחלקה עם ערכים לא תקינים, נוצרת נקודה עם ערכים תקינים רנדומליים במקום הערכים השגויים.
- בנאי ברירת מחדל של המחלקה יוצר נקודה רנדומלית בגבולות הנ"ל.
- למחלקה מתודה calcDistance(Point other) המחשבת מרחק בין הנקודה הנוכחית לנקודה נתונה (לפי נוסחת מרחק בין שתי נקודות במערכת צירים).

2. מחלקת Junction

- יורשת ממחלקת Point
- המחלקה מייצגת צומת דרכים.
- מחזיקה רשימות נפרדות לדרכים נכנסות ודרכים יוצאות.
- בכל צומת מוצב רמזור (ראו סעיף 3).

3. רמזורים.

עליכם לממש 3 מחלקות רמזורים בהתאם למה שלמדנו בכיתה ובמשימות מעבדה. לכל רמזור יש זמן השהייה (בפעימות). זה הזמן שאור נשאר ירוק לכביש מסוים. בכל החלפת אורות האור הירוק ניתן לכביש אחד בלבד מבין הכבישים <u>הנכנסים</u> של הצומת. קיימים 2 סוגים של רמזורים: רמזור רנדומלי ורמזור עקבי. רמזור רנדומלי בוחר את הכביש שיקבל את האור הירוק באופן רנדומלי, ורמזור עקבי מעביר את האור הירוק בין הכבישים הנכנסים של הצומת לפי הסדר בו הם מופיעים.
בכל פעימה רמזור מפעיל את המתודה check() ובודק, האם זמן השהייה הסתיים. במידה וכן, האור מתחלף בהתאם לסוג הרמזור (רנדומלי או עקבי). כל החלפת אורות מודפסת (ראו דוגמאות הדפסה).

- רמזור יודע באיזה צומת הוא מוצב.
- לרמזור מתודה getCurrentGreen המחזירה את הכביש בו יש אור ירוק ברגע זה.
- זמן השהייה (delay) הוא מספר רנדומלי בין 2 ל-4 כולל.

4. מחלקת Road

- מייצגת כביש המחבר בין שני צמתים (שני הצמתים חייבים להיות שונים).
- נוצרת ע"י קבלת צומת התחלה וצומת סיום, במידה ומדובר באותו צומת, צמות סיום מוחלף בצומת רנדומלי.
- במהלך היצירה הכביש הנוצר מתווסף לרשימת דרכים יוצאות של צומת ההתחלה ולרשימת דרכים נכנסות של צומת הסיום.

5. מחלקת Map

- מייצגת מפה המכילה צמתים ודרכים.
- בנאי אחד במחלקה מקבל מספר שלם, המייצג את כמות הצמתים הרצויה. הבנאי יוצר כמות זו של צמתים רנדומליים, ולאחר מכן מחבר ביניהם ע"י כבישים באופן רנדומלי גם כן. כל כביש הוא חד כיווני, כך ששני צמתים יכולים להיות מחוברים ע"י שני כבישים בעלי כיוון מנוגד.
לאחר סיום בניית הדרכים, בחלק מהצמתים שקיימים בהם כבישים נכנסים מתווספים רמזורים (הסתברות להוספת רמזור לצומת עם כבישים נכנסים היא 0.25). הבחירה של סוג הרמזור שמוצב בצומת נעשית באופן רנדומלי.
- בנאי נוסף מקבל אוסף של צמתים ואוסף של דרכים ואלה יהיו הצמתים והדרכים שבמפה. הבנאי מוסיף רמזורים לפי אותם כללים בהם משתמש הבנאי הקודם.

- המחלקה מכילה מתודה שבוחרת צומת רנדומלי מאוסף הצמתים שבמפה, ובונה מסלול רנדומלי המתחיל בצומת זה וממשיך ממנו עד שמגיע לצומת ללא מוצא (ללא כבישים יוצאים) או לאורך מקסימלי של 4 רחובות. המסלול מיוצג ע"י אוסף של כבישים, כאשר צומת סיום של כביש הוא צומת התחלה של כביש הבא במסלול.

.6 מחלקת Vehicle

- מייצגת כלי רכב.
- בנאי המחלקה מקבל מפה כארגומנט. עבור כל רכב מוקצה מסלול רנדומלי (המתואר במחלקת מפה). לכל רכב מהירות רנדומלית, כאשר ערכי המהירות יכולים לנוע בין 30 ל-120.
- לרכב מתודה ()move המקדמת את זמן הנסיעה ומדפיסה את מצבו ומיקומו הנוכחי של הרכב. בכל פעולת move נבדק האם הרכב סיים לנסוע בכביש הנוכחי (ע"י השוואה של אורך הכביש עם זמן נסיעה בכביש זה, מוכפל במהירות הרכב). במידה והרכב סיים את הכביש, הוא עובר לכביש הבא, אחרת ממשיך לנסוע בכביש הנוכחי. במידה וסיים את הכביש והכביש הוא כביש אחרון במסלול, מודפסת הודעה על כך שרכב סיים את המסלול והודעה זאת תודפס עבור רכב זה עד סוף המשחק.
- בשלב זה תנועת הרכבים אינה מושפעת מרמזורים.

.7 מחלקת DrivingGame

- המחלקה שמכילה את כל רכיבי המשחק ומתפעלת אותם.
- המשחק מתנהל בתורות (פעימות), כל תור מייצג יחידת זמן, למשל אם מהירות הרכב היא 60 אז בכל תור (פעימה) הרכב יתקדם ב-60 יחידות מרחק.
- בנאי המחלקה מקבל את כמות הצמתים הרצויה ואת כמות הרכבים. לאחר מכן יוצר מפה עם כמות צמתים זו ורכבים בכמות הנתונה.
- במחלקה מתודה play (int turns) המקבלת את כמות הפעימות הרצויה, ומריצה את המשחק למשך כמות פעימות זו. בכל פעימה כל אחד מהרכבים מבצע ()move וכל אחד מהרמזורים מבצע ()check.

.8 מחלקת Program

- נתונה בקובץ נפרד לצורך בדיקות עצמיות בלבד, אין צורך להגיש אותה.
- התוכנית כולה צריכה לפעול ע"י הרצת מחלקה זאת והפלט צריך להיות זהה (חוץ מערכים רנדומליים) לפלט שניתן בקובץ נפרד.

עבודה נעימה!