מונחה עצמים – הגשה 1
חורף 2020
203840095 - צחי פרץ
205836455 – שימי הללויה

שאלה 1

ב1.
כרגע הrequires מקל יותר , מכיוון שאנו מאפשרים למשתמש להזין קלט בו אחת מהנקודות ,p1 או p2 , חופפת לנקודת הסיום של המסלול .
המימוש ישתנה בכך שצטרך לבדוק האם הsegment החדש יקיים את התנאי הראשון או השני בדרישה החדשה – לפי בדיקה זו, נדע כיצד ליצור את ה newSegment החדש בהתאם לכיוון של p1 ו p2 .
ב2. המפרט החדש שנוצר עם דרישות מקלות יותר , לכן הוא חזק יותר .

ג.
אם בדרישות אנו לא מקבלים את הדרישה בה האורך של הסגמנט חייב להיות 0 , נצטרך להתייחס למצב הזה במימוש שלנו .
במתודת getHeading הורדנו את הrequoire.
תמכנו בקונסטרקטור שלנו במקרה בו קיבלנו מקטע באורך אפס , שם נחזיר זווית 0 .

ד.route הוא אכן subtype של geofeature – הוא מקיים את עיקרון ההחלפה של איסקו לפיו :

- We say B is a (true) subtype of A if B has a stronger specification than A
  - This is not the same as a Java subtype
  - Java subtypes that are not true subtypes are confusing & dangerous
    - But unfortunately common poor design

כלומר , ל route יש מפרט חזק יותר מgeofeature , חכן הוא subtype של geofeature .
ההפך לא נכון .

ה.
דוגמא למחלקה נוספת שתוכל לרשת מ routeFormmater , יכולה להיות הנחיות תנועה לרוכב אופניים .
נרצה לתת לרוכב האופניים הוראות חישוב כיוונים בצורה זהה , כפי שמוגדר במחלקת האב routeFormmater . במחלקת הבן , שהיא ההוראות לרכיבה , נממש את computeLine בצורה כזו , שנחשב עבור רוכב האופניים כמה זמן ייקח לעבור כל מקטע.

דוגמא למחלקה שלא תוכל לרשת מ routeFormmater
נשים לב שלמחלקה האבסטרקטית routeFormmater , יש 3 מתודות :

```
public String computeDirections(Route route, double heading) {

public abstract String computeLine(GeoFeature geoFeature, double origHeading);

protected String getTurnString(double origHeading, double newHeading) {
```

אחד מעקרונות ההורשה הוא שמילות הגישה של מחלקת הבן יכולות להיות מחמירות פחות, אך לא להפך . לדוגמא , מצב בו מחלקת public של האב (לדוגמא computedirections) להיות protected

אצל הבן הוא בלתי אפשרי .(לדוגמא כאשר אנו מבצעים override לcomputedirections במחלקת הבן)

<u>שאלה 2</u>

ב. בגלל ששדה הroute הוא immutable, בכל הוספה של סגמנט לדרך אנחנו יוצרים instance חדש למחלקת route , כלומר אנחנו משכפלים את הinsatace הקודם ומשרשרים אליו את הסגמנט החדש .
כאשר מספר האיברים בroute גדול , השכפול דורש משאבים רבים של זיכרון ושל חישוב .
דרך מימוש חלופית תשתמש בroute שהוא mutable. כל בכל הוספת סגמנט , נוסיף אותו לסוף ה route הנוכחי , ללא צורך בשכפול.

ג.
הפרמטרים שהוא מקבל הם :
א. frame owner – הוא החלון הראשי של ה gui. מכיוון שמימשנו דיאלוג , צריך להגדיר למי הדיאלוג מגיב , כלומר מי יפתח אותו.
ב. pnlparent – הוא instance של המחלקה שהשדות שלה מופיעות בgui הראשי ,אנחנו משתמשים בה כדי לשנות את השדות בחלון ה dialog שנפתח לנו – כלומר , בעת לחיצה על add, אנו מוסיפים סגמנט לroute של ה pnlparent.

ד.
מבחינת ה dialog נוסיף כפתור remove ונממש בו handler שיימחק את ה segment מהroute .
נצטרך לממש מתודת remove ב route (שייתמוך גם במצב בו ביקשנו לעשות remove לסגמנט שלא קיים ב route). נצטרך גם לתמוך במימוש שמבקש לקטוע את הroute, לדומגא המסלול הוא {א-ב-ג-ד-ה} ומחקנו את ג׳ – נקבל שני מסלולים שהם : {א ,ב} ,{ד,ה}

<u>שאלה 3</u>

Abstraction function :
Poly class represent a polynomial function with integer coeffienets .
Terms is a list of PolyTerms (which contain coeff and power of x).
Each elemnt in the PolyTerms list has its coeff membet and power member .

b. Representation invariant :
Terms != null.
Terms.size > 0
For each e in Terms :
```
   e != null
   e.power >= 0 .
   if ( e.power != 0)
      e.coeff !=0
   else
      e.coeff = 0
```

```
for i in in range(terms.size-1) :
   terms[i].power <= terms.[size-1].power
```

c. new Representation invariant will require that , in place i in the terms list will be only the power i of x.
if the polynom doesn't contains increment sequence of integers ,we would still add element I no to place I , and place 0 between. For example ,if the polyon was (x+x^3) . our termsList will be [(0,0) , (1,1) , (0,2) ,(1,3)] // (coeff, power).
In that case , in coeff function , we will get the coeff of power I , by accesing it directly .
:

```
Terms != null.
Terms.size > 0
For each e in Terms :
    e != null
    e.power >= 0 .
    if ( e.power != 0)
        e.coeff !=0
    else
        e.coeff = 0

for i  in in range(terms.size-1) :
   terms[i].power == i
```