שלב 1 - הגדרת פרימיטיבים וגופים גיאומטריים

<u>על השלב חלים כל הכללים המופיעים במסמך "הנחיות כלליות לעבודה על שלבי הפרויקט בקורס"</u>

<u>בשלב ראשון של הפרויקט</u> נגדיר אוסף של מחלקות בעזרתם נוכל לתאר סצנה גרפית בהמשך.

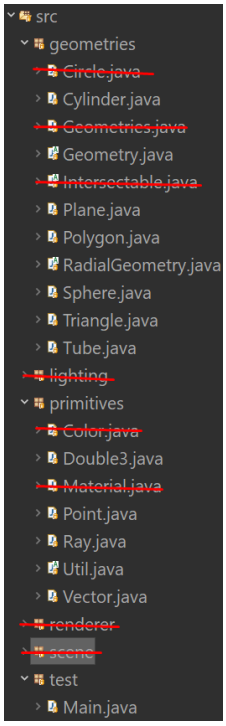
1. **נפתח שלוש חבילות הנדרשות שתי המודולים שקיבלתם לתרגיל:**
   - **primitives** (עבור מחלקות הייסוד לישויות של גאומטריה)
   - **geometries** (לגופים גאומטריים)
   - **test** (לתוכנית הראשית לבדיקת שלב 1)

   נ.ב. שימו לב שעבור כל חבילה חדשה נפתחת אוטומטית תת-תיקיה בשם זהה בתיקיית קוד מקור של הפרויקט (**src**)

2. **נוסיף את המודולים שקיבלתם בתיבת הגשת השלב:**
   - **Util.java** בחבילה (תיקייה) **primitives**
   - **Double3.java** בחבילה (תיקייה) **primitives**
   - **Polygon.java** בחבילה (תיקייה) **geometries**
   - **Main.java** בחבילה (תיקייה) **test**

נ.ב. *חובה לעבור על המודולים האלה שקיבלתם – על מנת ללמוד איך לכתוב ולעצב את הקוד שלכם ואיך לכתוב את התיעוד (במיוחד javadoc) בהמשך*

נ.ב. אם צצה לכם בעיה של מידע של מודול לא תקין – זה אומר שבהקמת הפרוייקט לא ביטלתם סימון הוספה של הקובץ **module-info.java**: הפתרון – למחוק את הקובץ הזה מהפרויקט.




```
<terminated> Main [Java Application] C:\java\bin\javaw.exe (21 21:36:27 – 21:36:23 ,2022 נפבר)
Error occurred during initialization of boot layer
java.lang.module.FindException: Error reading module: C:\Users\[redacted]
Caused by: java.lang.module.InvalidModuleDescriptorException: Main.class found in top-lev
```

יחד עם זאת, מי שחושב שיכול ללמוד ולהשתלט על עבודה עם מודולים של java, יוכל להמשיך לעבוד עם הקובץ הזה – רק יצטרך למצוא איך הוא פותר את הבעיה (רמז: יש להוסיף בקובץ הזה את החבילות [packages] שהוספנו)

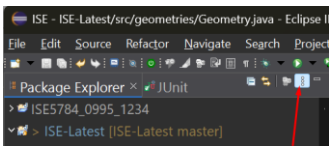
```
exports primitives;
exports geometries;
```

ובכל מקרה האחריות על עבודה בפרויקט עם תמיכה במודולים – על הסטודנט בלבד, ולא תבוא לו ישועה ממרצים

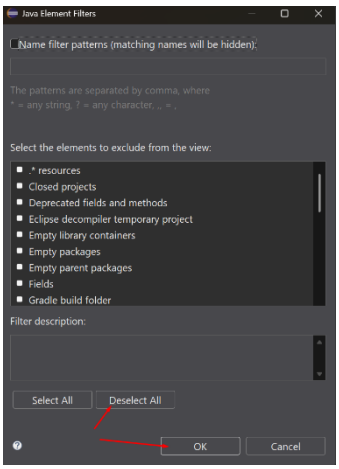
משמאל מופיע צילום מסך עם דוגמה (ב-eclipse) לתצוגת תוכן הפרויקט לאחר השלב הזה (כל מה מועבר עליו קו אדום – לא שייך לשלב הזה, נתעלם מזה)

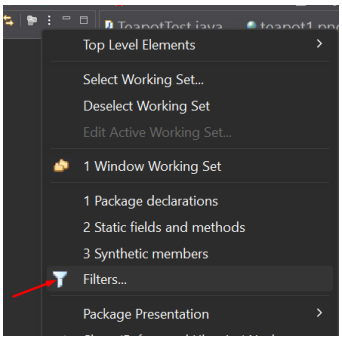
משתמשי eclipse – שימו לב שכדאי לעשות את שני השינויים הבאים בתצוגה בלשוניות Package Explorer ו-Project Explocrer:

- **בשתי ההגדרות מתחילים מלחיצ על שלושת הנקודות בצד ימין של הלשונית:**

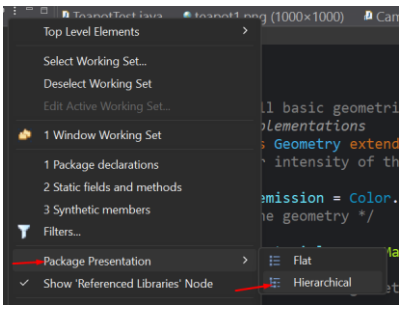


- **בראשונה (ביטול סינונים), בוחרים Filters... בתפריט שנפתחת, ובלון שייפתח – מבטלים את כל הסינונים (Deselect All):**

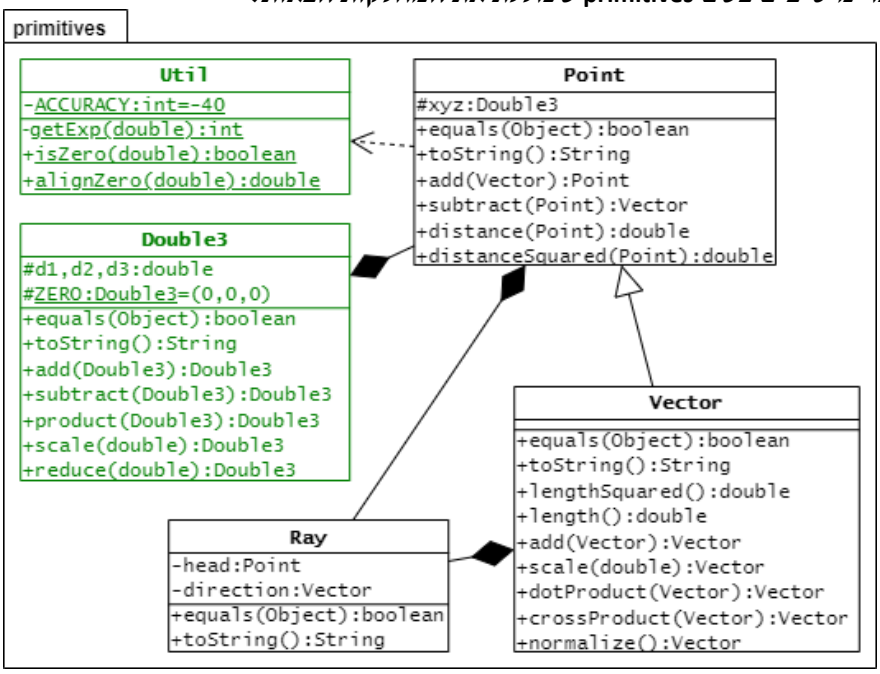




- **ובשנייה, משנים את התצוגה לתצוגה היררכית:**



3. נגדיר חבילה של פרימיטיבים בשם primitives שכוללת את המחלקות הבאות:



- **Util** (מחלקת שירות וכלי עזר) - לטיפול בדיוק החישובים, המודול ניתן בתוך תיבת ההגשה, **אסור לשנותו**
- **Double3** (שלשת מספרים דצימליים) , המודול ניתן בתוך תיבת ההגשה, **אסור לשנותו**
- **Point** (נקודה במרחב) - אובייקט ייסוד בגאומטריה של תלת ממד – נקודה בעלת 3 קואורדינטות, מכילה שדה של אובייקט של שלשת המספרים (שדה מטיפוס **Double3** – ע"י הכלה)
- **Vector** (וקטור) - אובייקט ייסוד בגאומטריה של תלת ממד, בעל כיוון וגודל, לפי הגישה של אלגברה לינארית – וקטור זה "מעין נקודה", מוגדר ע"י נקודת הסוף (יחסית למרכז הקואורדינטות), המחלקה **יורשת** ממחלקת **Point**
- **Ray** (קרן) - אובייקט ייסוד בגאומטריה של תלת ממד, המקום הגיאומטרי של כל הנקודות על ישר שנמצאות בצד אחד (המוגדר ע"י וקטור הכיוון של הקרן שחייב להישמר כ**וקטור יחידה** – וקטור מנורמל) מנקודה נתונה שנקראת ראש הקרן

היחס בין המחלקות כמו שמופיע בתרשים הנ"ל, ע"י הכלה וירושה בהתאם. כל המחלקות לא ניתנות לשינוי (**immutable**).

בכל מחלקה נוסיף:

- השדות בהתאם למופיע בתרשים לעיל
  - כל השדות עם מילה שמורה **final** על מנת שכל המחלקות תהיינה לא ניתנות לשינוי (**immutable**)
  - הרשאות גישה כפי שמופיע בתרשים!
- בנאים **constructors** (עם פרמטרים מתאימים) – ראו פירוט בהמשך.
  - **בכל מקרה אסור להגדיר בנאי ברירת מחדל ובנאי "העתקה"!**
- **אין** להגדיר מתודות אחזור (**get**) בשלב הזה
- **אסור** להגדיר מתודות עדכון (**set**) **בכלל**!
- נדרוש את המתודה **equals** ע"פ הכללים שנלמדו ותוך שמירה על העיקרון **DRY**
  - **חובה** להתבסס על **equals** של האובייקטים המוכלים
  - **חובה** להקפיד על התבנית הבאה (ולא על מתודה מחוללת ע"י כלי הפיתוח!):

```
package primitives;

/**
 * Class ClassName is the basic class representing a … of Euclidean geometry in Cartesian
 * 3-Dimensional coordinate system.
 * @author Student1 and Student2
*/
public class ClassName {
…
   @Override
   public boolean equals(Object obj) {
      if (this == obj) return true;
      return (obj instanceof ClassName other)
         && this.field1.equals(other.field1)
         && this.field2.equals(other.field2) …;
   }
}
```

- נדרוש את המתודה **toString** על מנת לאפשר **תצוגת** האובייקט – בצורה תמציתית אך נוחה לשימוש
  - **הדפסה אסורה!!!**
  - שימו לב בתצוגת נקודה ווקטור – יהיו לנו אובייקטים עם מספר נקודות, בכל נקודה יש שלוש קואורדינטות!
  - **חובה** להתבסס על מה שכבר קיים (לשמור על העיקרון **DRY**)

- במימוש המתודות של מחלקות הנקודה והווקטור **חובה** להקפיד על העיקרון **DRY**, בין היתר ע"י האצלת הפעולות למתודות של **Double3**
  - אך אין להשתמש במתודות שמייצרות אובייקט חדש **מיותר** לשימוש זמני בלבד – למשל אין להשתמש במתודת **product** של **Double3** בתוך מתודת **dotProduct** של מחלקת **Vector**
- באופן כללי **אין** ליצור **אובייקטים זמניים** לצורך חישובים (למשל לא לעשות חיסור נקודות על מנת לקבל מרחק ביניהן מאורך הווקטור הזמני שנוצר)
  - ***הסבר**: יצירת אובייקט חדש זו פעולה יקרה מבחינת זמן מעבד (ביצועים) ובנוסף להפעלת הבנאי כולל הקצאת זיכרון, כל המתודות שאתם מממשים עכשיו במחלקות של נקודה ושל וקטור – בסופו של דבר תזומנה בכמות אדירה ואתם לא רוצים לגרום לעצמכם להמתין ליצירת התמונה שעות במקום דקות*
  - ***הבהרה** לטעות נפוצה של סטודנטים: תזכרו ש**(משתנה ≠ אובייקט)***
  - ***הבהרה** נוספת בעקבו ריבוי טעויות: אין ליצור משתנה זמני שכל תפקידו הוא להחזיר ממנו את הערך שחישבו – תחשבו ישירות בתוך פקודת **return***
  - ***הבהרה** בהמשך להבהרה הקודמת – אם יש חישוב שחוזר על עצמו – בוודאי יש להגדיר משתנה עבור שמירת תוצאת החישוב הזה*
- הבנאי של **וקטור חייב** לזרוק חריגה במקרה של וקטור אפס, אסור להשתמש בחריגת בסיס **Exception** אלא בחריגה הרשומה בהמשך בתיאור של מחלקת וקטור
- הבנאי של **קרן חייב** להבטיח שווקטור הכיוון מנורמל
- **חובה** ליצור תיעוד בפורמט **javadoc** **לפני כל מחלקה ולפני כל איבר במחלקה** (שדה או מתודה)
  - **אין** לתעד את דריסות מתודות (**@Override**)
  - התיעוד בפורמט **javadoc** הנו תיעוד חיצוני ואסור שיתייחס לפרטי המימוש של המחלקה\המתודה

לכל מחלקה נוסיף מתודות שמאפשרות עבודה עם אובייקטים מטיפוס מתאים (אף מתודה למשנה את האובייקט שלה):

- **שירות (Util): המחלקה ניתנת באתר moodle**
- **שלישיית מספרים (Double3): המחלקה ניתנת באתר moodle**
- **נקודה במרחב (Point):**
  - שדה של ערכי הקואורדינטות מטיפוס **Double3**, עם הרשאת גישה **protected**
  - שדה סטטי קבוע מטיפוס **Point** בשם **ZERO** ועם הרשאה **public**, השדה מאותחל עם נקודה של מרכז מערכת הקואורדינטות (ראשית הצירים)
  - בנאי שמקבל שלושה מספרים מטיפוס **double** עבור ערכי הקואורדינטות, עם הרשאת גישה **public**
  - בנאי שמקבל אובייקט מטיפוס **Double3**, עם הרשאת גישה **public**
    נ.ב. אסור ליצור בנאי ברירת מחדל ובנאי העתקה
  - במחלקה **לא תהינה** מתודות מאחזרות (**get**) בשלב הזה
  - **subtract** – חיסור וקטורי – מקבלת נקודה שניה בפרמטר, מחזירה וקטור מהנקודה השנייה לנקודה שעליה מתבצעת הפעולה
  - **add** – הוספת וקטור לנקודה – מחזירה נקודה חדשה
    נ.ב. בשתי המתודות הנ"ל יש להשתמש במתודות המתאימות של **Double3**
  - **distanceSquared** – המרחק בין שתי נקודות בריבוע (אמנם לא מופיעה בארכיטקטורה לעיל אך חובה להגדירה ולממשה)
  - **distance** – מרחק בין 2 נקודות (במימוש יש להשתמש במתודה הקודמת **distanceSquared**)
    - **אסור** להשתמש במתודה **Math.pow()** על מנת לחשב ערך בריבוע
      - המתודה הנ"ל מחשבת גם חזקות עם מעריכים ממשיים בעזרת חישובים מקורבים ע"י טורים מתמטיים, לכן לא כדאי להשתמש בה – אנחנו הולכים לעשות המון חישובים, תכפילו את הערך בעצמו!
    - **אסור** ליצור מתודות נוספות אלא אם השתכנעתם שאתם חייבים אותן ויש להן משמעות גאומטרית או אלגברית!
- **וקטור** (**Vector**) (בכל המתודות יש להשתמש בשיטה של **אלגברה לינארית** ולא בשיטה טריגונומטרית):
  - **יורש מנקודה** (כפי שלמדתם באלגברה לינארית – וקטור זה "סוג של נקודה")
  - בנאי המקבל: שלושה מספרים מטיסוס **double**, עם הרשאת גישה **public**
  - בנאי המקבל אובייקט מטיפוס **Double3**, עם הרשאת גישה **public**
    נ.ב. אסור ליצור בנאי ברירת מחדל ובנאי העתקה
  - **כל** הבנאים "יזרקו" חריגה מטיפוס **IllegalArgumentException** עם טקסט מתאים במקרה של "ווקטור אפס", הבדיקה תתבצע ע"י השוואת שדה הקואורדינטות לקבוע **ZERO** שמוגדר במחלקה **Double3**, בעזרת המתודה **equals**
  - **add** – חיבור וקטורי (מחזירה וקטור **חדש**)
    - מתודת החיסור לא נדרשת – היא כבר קיימת במחלקת האב (נקודה) ומתאימה גם לווקטור
  - **scale** – מכפלת ווקטור במספר - סקלר (מחזירה וקטור **חדש**)
  - **נ.ב.** בשתי המתודות הנ"ל יש להשתמש במתודות המתאימות של **Double3**
  - **dotProduct** – מכפלה סקלרית (**dot-product**) – יש להשתמש בנוסחה של אלגברה לינארית ולא טריגונומטרית, אין להשתמש במתודות **Double3** אלא לבצע חישוב ערכים "גולמי"
  - **crossProduct** – מכפלה וקטורית – מחזירה וקטור **חדש** שניצב לשני הווקטורים הקיימים (**cross-product**), יש להשתמש בנוסחה של אלגברה לינארית ולא טריגונומטרית

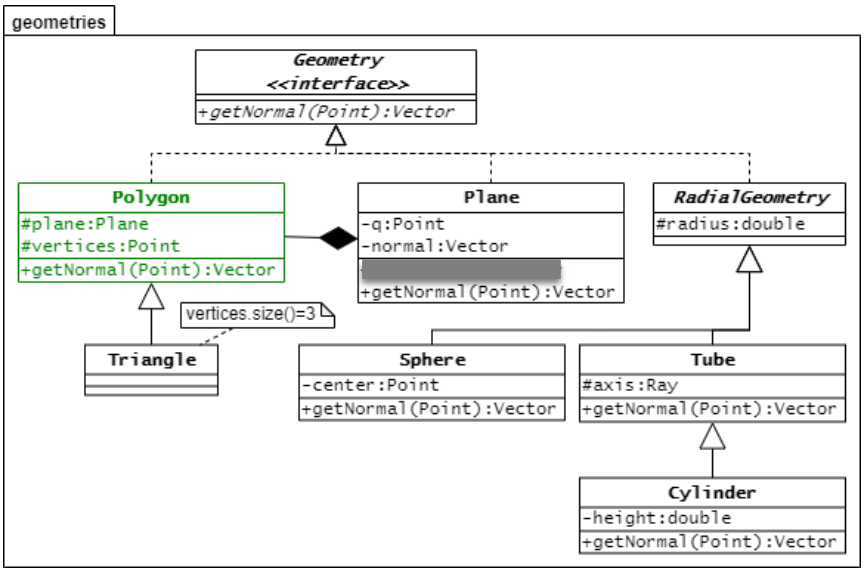
- **lengthSquared** – חישוב אורך הווקטור בריבוע
  - **אין** להשתמש בחישוב מרחק בין ראש הווקטור למרכז קואורדינטות (תוך יצירת אובייקט זמני), ניתן להשתמש בתכונה מתמטית: $\vec{v} \cdot \vec{v} = \vec{v}^2 = |\vec{v}|^2$
- **length** – חישוב אורך הווקטור (יש להשתמש במתודה הקודמת **lengthSquared**)
- **normalize** – מתודת נרמול שמחזירה וקטור **חדש** מנורמל – וקטור יחידה באותו כיוון כמו הווקטור המקורי
- **קרן (Ray)** (מעין ״חצי-ישר״, בהמשך ייצג גם לפעמים ישר או קטע):
  - בנאי שמקבל נקודה ווקטור, עם הרשאת גישה **public**
    נ.ב. אסור ליצור בנאי בררת מחדל ובנאי העתקה
  - וקטור שמתקבל לבנאי לא חייב להיות מנורמל, אך וקטור הנשמר בקרן חייב להיות מנורמל: ***לא לשכוח לנרמל את הווקטור לפני שמירה***

בעזרת חבילת הפרימיטיבים שהגדרנו נוכל עכשיו להגדיר אוסף של מחלקות בעזרתם נוכל לתאר צורות וגופים הנדסיים (גיאומטריים).

המשך בעמוד הבא

4. בדומה למה שעשינו קודם נוסיף חבילת הגופים בשם geometries שמכילה אוסף של מחלקות לתיאור גופים גאומטריים. כדי לאפשר לבנות סצנה מאוסף של גופים שונים נבנה את החבילה בצורה היררכית.

נגדיר **מחלקה אבסטרקטית** בשם **Geometry** עבור גוף גאומטרי כלשהו ואחר כך נגדיר מחלקות עבור הגופים השונים לפי הארכיטקטורה הבאה:



נ.ב. אמנם **Geometry** מופיע בתרשים כממשק (**interface**), אך ידוע לכם מהקורס התיאורטי שהיא תהפוך בהמשך למחלקה אבסטרקטית. לכן נגדיר אותה כך כבר עכשיו (וגם מודול ה-**Polygon** שקיבלם – משתמש בה בירושה ולא במימוש).

נבחר במספר גופים גאומטריים למימוש בפרויקט (ברור שניתן להרחיב את הפרויקט לגופים רבים נוספים). כל המחלקות של הגופים האלה:

- כולן יהיו "לא ניתנות לשינוי" (**immutable**), זאת אומרת – כל השדות יהיו עם מילה שמורה **final**
- הרשאות גישה של כל השדות: במחלקות מורישות – **protected**, ובמחלקות שאינן מורישות – **private**
- **כל מחלקה (למעט מישור) תכלול בנאי אחד בלבד** המקבל פרמטרים לאתחול שדות המחלקה המתאימים (אסור לאתחל שדות מחלקת אב אלא רק דרך הפעלת בנאי מתאים של מחלקת האב)
  - **אסור** להגדיר בנאי בררת מחדל ו-\או בנאי העתקה
  - **למחלקת מישור יהיו שני בנאים** – ראו בהמשך
- **אסור** להגדיר מתודות אחזור (**getter**) ומתודות עדכון (**setters**)
- **לא** נדרוס את המתודה **toString** (מותר לדרוס אותה אם אתם רוצים)
- **אסור** לדרוס את המתודה **equals**
- **חובה** ליצור תיעוד בפורמט **javadoc** **לפני כל מחלקה ולפני כל איבר במחלקה**, **למעט** דריסת מתודות (**@Override**)
  - התיעוד בפורמט **javadoc** הנו תיעוד חיצוני ואסור שיתייחס לפרטי המימוש של המחלקה\המתודה

מחלקות אבסטרקטיות:

- נוסיף למחלקה האבסטרקטית **Geometry** הצהרת מתודה בשם **getNormal** המקבלת פרמטר אחד מטיפוס נקודה [על פני הגוף הגאומטרי] ומחזירה ווקטור הנורמל (האנך) לגוף בנקודה זו. נוסיף תיעוד **javadoc** למחלקה ולמתודה.
- נגדיר מחלקה אבסטרקטית **RadialGeometry** הממשת את הממשק **Geometry**, המחלקה תכלול:
  - שדה **radius**
  - בנאי שמקבל בפרמטר ערך עבור הרדיוס ומאתחל את השדה בהתאם
  - כדאי להוסיף שדה עבור **רדיוס בריבוע**, ולאתחל אותו (לחשב) בתוך הבנאי הנ"ל – השדה יהיה שימושי בשלב 3

גופים שטוחים:

- מחלקה של מישור – **Plane**, המחלקה תכלול:
  - שדות עבור נקודה ווקטור הנורמל
  - שני בנאים
    - בנאי המקבל בפרמטרים 3 נקודות
      - הבנאי אמור לחשב את הנורמל לפי מה שנלמד על נורמל למשולש – בשלב הזה יישמר ערך **null** בשדה הנורמל (המימוש המלא יתבצע בשלב הבא), כמו כן הבנאי ישמור את אחת הנקודות כנקודת הייחוס של המישור
    - בנאי המקבל בפרמטרים נקודה ווקטור הנורמל
      - יש לממש את הבנאי הזה כראוי
      - שימו לב שהפרמטר של וקטור לא חייב להיות מנורמל, אך חובה לשמור את הנורמל כמנורמל
- מחלקה של מצולע – **Polygon**, המחלקה מצורפת בתיבת ההגשה במלואה לפי השלב הזה

- **בשלב הזה אסור לעשות כל שינוי בתוך המודול הזה!**
- המחלקה תכיל אוסף (בלתי ניתן לשינוי - **immutable**) ממוספר של קודקודי המצולע:

```
List<Point> vertices;
```

- כמו כן החלקה תכיל את [הפניה ל-] המישור שהמצולע נמצא בו
- הבנאי יקבל כמות מתאימה של נקודות הקודקוד של פוליגון ע״פ הסדר שלהם מבחינת חיבור ביניהם ע״י צלעות המצולע ויבדוק האם הרשימה של הקודקודים של המצולע תקינה (מסודרת לפי חיבורי הצלעות, אין קודקודים על אותו ישר, המצולע קמור) [הקוד מבוסס על גרסה מתקדמת **Java**]
  - השימוש בפרמטר **vertices** הנ״ל בתוך הבנאי הוא כמו במערך של נקודות – **vertices[i]**
- הבנאי מוודא שיש לפחות 3 קודקודים (אחרת תיזרק חריגה מטיפוס **IllegalArgumentException**)
- הבנאי יוצר מישור ע״י שלושת הנקודות הראשונות

נ.ב. אם המודול לא מתקמפל אצלכם, עשויה להיות אחת משלושת הסיבות הבאות לכך:

1. לא התקנתם את ה-**JDK** כפי שנדרש לפי ההנחיות בקורס ויש לכם במחשב גרסת **Java** ישנה (מאד!)
2. עוד לא הגדרתם ממשק **Geometry** ו-\או מחלקת מישור (**Plane**)
3. לא הגדרתם מתודה מאחזרת בסיסית (ללא פרמטרים) במחלקת המישור

- מחלקה של משולש (מצולע בעל קודקודים) [**Triangle**]
  - יורש מהמצולע, וללא שדות

גופים תלת ממדיים:

- בכל המחלקות יש להוסיף בנאי (עם הרשאה **public**) המקבל והמאתחל את כל הנתונים הרלוונטיים של הגופים
- מחלקה של כדור (מכיל שדות של נקודת המרכז ורדיוס) – **Sphere**
- מחלקה של צינור (מכיל שדות של רדיוס וקרן של הציר הראשי) – **Tube**
- מחלקה של גליל (מכיל שדה של גובה) – **Cylinder**

כל המימושים של מתודה getNormal (לפי התרשים) של הממשק **Geometry** בשלב הזה:

- יחזירו **null**
- **למעט במישור** שיחזיר את ערך הנורמל שלו
- **למעט בפוליגון** (מצולע) שכבר מחזיר את הנורמל ע״י האצלה למישור המוכל בו

תוכנית ראשית שבודקת את נכונות הפעולות מצורפת לתרגיל. יש לוודא שהבדיקות מסתיימות בהצלחה – זאת אומרת שלא מודפסת שום שורה שמתחילה במילה **ERROR** ואין חריגות.

ציון התרגיל יינתן:

- לפי אחוז הכיסוי (הביצוע) של משימת התרגיל – כולל בדיקות כל הפעולות
- על ההקפדה על הכללים (פרמוט [הזחות, רווחים, שורות רווח]
- על תיעוד **javadoc** למחלקות ולמתודות
- על הקפדה על מוסכמות שמות
- על יעילות ביצוע והקפדה על עקרונות דיזיין
- על הקפדה בדרישות\תבנית מימושים של דריסות המתודות **equals** ו-**toString**

חובה לבצע יצירת תיעוד **Javadoc** (**Generate Javadoc**) עבור כל הפרויקט, מהרשאה **private**, **לעקוב** אחרי כל התקלות ביצירת התיעוד **ולתקן** אותן. הנחיות יצירת התיעוד מופיעות בהחיות הכלליות לעבודה על כל השלבים.

שימו לב – אי הקפדה של ההנחיות לעיל עלולה לגרור הורדה בציון