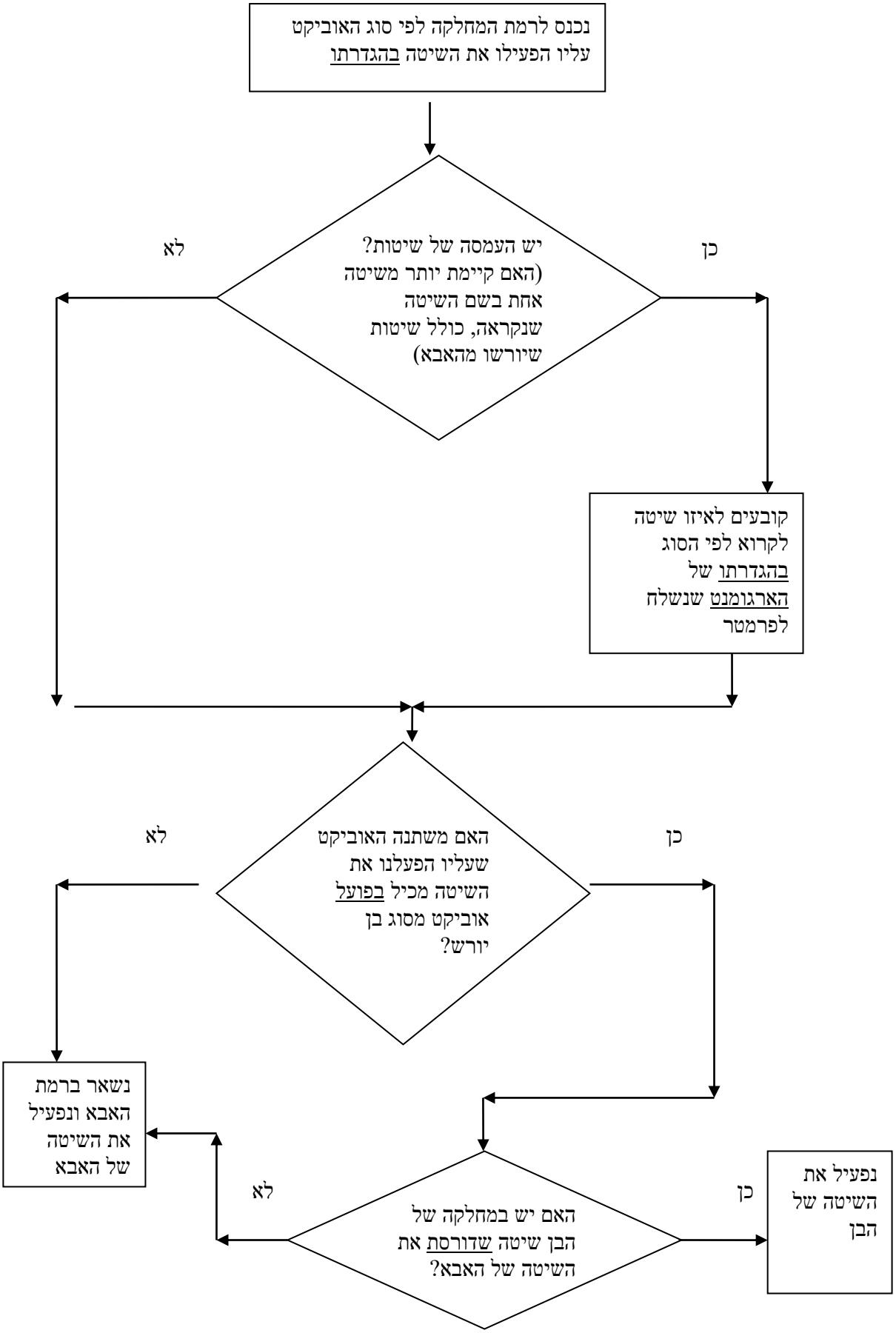
העמסה ודריסה

הסבר מושגים בהם נשתמש בהסבר:

אם Kid היא מחלקה שיורשת ממחלקה בשם Dad, ובהינתן ההגדרה הבאה:
Dad d=new Kid()
נאמר שהאובייקט d הוא לפי ההגדרה מסוג המחלקה Dad אך בפועל מכיל אובייקט מסוג Kid.

העמסה overloading - מצב שבו יש במחלקה יותר משיטה אחת בעלת אותו שם (השיטות חיבות לקבל פרמטרים שונים).

דריסה overriding מצב שבו יש במחלקה של הבן היורש שיטה בעלת אותה חתימה של שיטה במחלקה של האבא אך הקוד נכתב מחדש.


נכנס לרמת המחלקה לפי סוג האוביקט עליו הפעילו את השיטה בהגדרתו
יש העמסה של שיטות? (האם קיימת יותר משיטה אחת בשם השיטה שנקראה, כולל שיטות שיורשו מהאבא)
כן
לא
קובעים לאיזו שיטה לקרוא לפי הסוג בהגדרתו של הארגומנט שנשלח לפרמטר
האם משתנה האוביקט שעליו הפעלנו את השיטה מכיל בפועל אוביקט מסוג בן יורש?
כן
לא
נשאר ברמת האבא ונפעיל את השיטה של האבא
האם יש במחלקה של הבן שיטה שדורסת את השיטה של האבא?
כן
לא
נפעיל את השיטה של הבן

ג'ודי אייזקס

```
public class Dad
{
  public void method( Dad d){
      System.out.println("Dad's method");
   }
}
public class Kid extends Dad
{
 public void method(Kid k){
     System.out.println("in kids method overloading dad's method");
   }

   public void method (Dad d){
      System.out.println("in kid's method, overriding dad's method");
   }
}
public class Test
{
  public static void main(String []args){
          Dad dad=new Dad();
          Dad reallyKid=new Kid();
          Kid kid=new Kid();

          dad.method( dad);
          dad.method(reallyKid);
          dad.method(kid);
          reallyKid.method(dad);
          reallyKid.method(reallyKid);
          reallyKid.method(kid);
          kid.method(dad);
          kid.method(reallyKid);
          kid.method(kid);
        }
      }
```

	בפועל	לפי ההגדרה
dad	Dad	Dad
reallyKid	Kid	Dad
kid	Kid	Kid

output: הסבר:

in Dad's method בשלשלת הקריאות הראשונות השיטה מופעלת על אובייקט מסוג האבא והוא
in Dad's method נשאר ברמת האבא
in Dad's method ללא קשר לפרמטר שקבלה השיטה

in kid's method, overriding dad's method בשלושת הקריאות האילו השיטת מופעלת על אובייקט
in kid's method, overriding dad's method מסוג אבא שבפועל בזמן ריצה הוא בן, ולכן בגלל הדריסה וה-
in kid's method, overriding dad's method dynamic binding
ירד לרמה של הבן לשיטה הדורסת ללא קשר לפרמטר שקבלה השיטה

in kid's method, overriding dad's method בשלושת הקריאות האלו השיטה מופעלת על
in kid's method, overriding dad's method אובייקט מסוג הבן, שם יש 2 שיטות,
in kids method overloading dad's method overloading
הבחירה לאיזו שיטה ללכת הוא לפי סוג
הפרמטר בהגדרתו (בקומפיילר) ללא קשר למה שהוא בפועל בזמן ריצה

```
public class Dad
{
   public void method( Dad d){
       System.out.println("Dad's method");
    }
}
public class Kid extends Dad
{
  public void method(Kid k){
      System.out.println("in kids method overloading dad's method");
    }
   //הורדנו את השיטה הדורסת
}
public class Test
{
   public static void main(String []args){
            Dad dad=new Dad();
            Dad reallyKid=new Kid();
            Kid kid=new Kid();

            dad.method( dad);
            dad.method(reallyKid);
            dad.method(kid);
            reallyKid.method(dad);
            reallyKid.method(reallyKid);
            reallyKid.method(kid);
            kid.method(dad);
            kid.method(reallyKid);
            kid.method(kid);
          }
       }
```

output: הסבר:

in Dad's method בשלשלת הקריאות הראשונות השיטה מופעלת על אובייקט מסוג האבא והוא

in Dad's method נשאר ברמת האבא

in Dad's method <u>ללא קשר לפרמטר שקבלה השיטה</u>

in Dad's method בשלושת הקריאות האילו השיטה מופעלת על אובייקט

in Dad's method מסוג אבא <u>שבפועל בזמן ריצה</u> הוא בן, מכיוון שאין הדריסה של השיטה אצל הבן

in Dad's method יישאר ברמת האבא ללא קשר לפרמטר שקבלה השיטה

in Dad's method בשלושת הקריאות האלו השיטה מופעלת על

in Dad's method אובייקט מסוג הבן, שם יש 2 שיטות,

in kids method overloading dad's method overloading

השיטה שירש מהאבא והשיטה של הבן

הבחירה לאיזו שיטה ללכת הוא לפי סוג

<u>הפרמטר</u> בהגדרתו (בקומפיילר) ללא קשר למה שהוא בפועל בזמן ריצה static binding

```
public class Dad
{
  public void method(Kid k){ //  הוספנו שיטה לאבא
     System.out.println("in dads new method overloading dad's method");
   }
  public void method( Dad d){
      System.out.println("Dad's method");
   }
}
public class Kid extends Dad
{
 public void method(Kid k){
     System.out.println("in kids method overloading dad's method");
   }

   public void method (Dad d){
      System.out.println("in kid's method, overriding dad's method");
   }
}
public class Test{
  public static void main(String []args){
            Dad dad=new Dad();
            Dad reallyKid=new Kid();
            Kid kid=new Kid();
            dad.method( dad);
            dad.method(reallyKid);
            dad.method(kid);
            reallyKid.method(dad);
            reallyKid.method(reallyKid);
            reallyKid.method(kid);
            kid.method(dad);
            kid.method(reallyKid);
            kid.method(kid);
          }
       }
```

Dad's method בשלשלת הקריאות הראשונות השיטה מופעלת על אובייקט מסוג האבא והוא
Dad's method נשאר ברמת האבא
in dads new method overloading dad's method מיכיוון שיש העמסה, נקבע לאיזו שיטה ללכת לפי ההגדרה של הפרמטר

בשלושת הקריאות האילו השיטה מופעלת על אובייקט שבפועל בזמן ריצה הוא בן, אך לפי ההגדרה הוא אבא. נכנסים ברמה של האבא, ומיכיוון שיש העמסה, נקבע לאיזו שיטה ללכת לפי ההגדרה של הפרמטר ואח"כ בגלל שבפועל יושב אוביקט מסוג הבן בגלל הדריסה וה - dynamic binding ירד לרמה של הבן לשיטה הדורסת

in kid's method, overriding dad's method
in kid's method, overriding dad's method
in kids method overloading dad's method

in kid's method, overriding dad's method בשלושת הקריאות האלו השיטה מופעלת על
in kid's method, overriding dad's method אובייקט מסוג הבן, שם יש 2 שיטות
in kids method overloading dad's method הבחירה לאיזו שיטה ללכת הוא לפי סוג הפרמטר בהגדרתו (בקומפיילר) ללא קשר למה שהוא בפועל בזמן ריצה