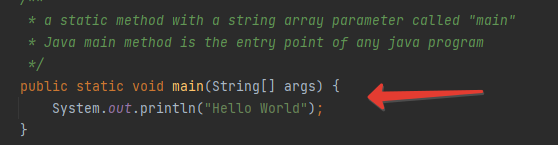
**The Java Program Entry Point**

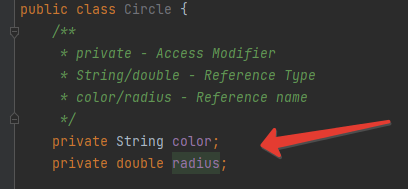


לכל תוכנה שכתובה בג'אווה תהיה שיטה סטטית הנקראת main ממנה התוכנה תתחיל, שיטה זו תקבל כפרמטר מערך של מחרוזות. בהמשך נראה מה פירוש הדברים הללו.

בדוגמא לעיל: התוכנה תדפיס לקונסולה “Hello World”

**Java Class Structure**

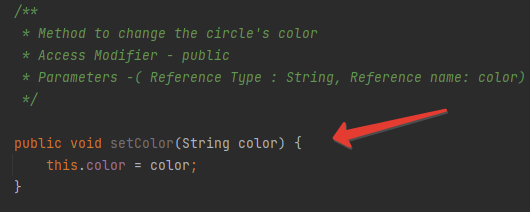
Class Properties



את התכונות של המחלקה בדרך כלל נגדיר בראש המחלקה, לכל תכונה יש משני גישה כדי שנוכל להגדיר למי יש גישה לתכונות אלה ובאיזה מצבים. תכונות המחלקה הם אובייקטים ובפרט לכל אובייקט יש את סוגו (לדוגמא: מספר,מילה,אות,מספר עשרוני).

לכל תכונה תהיה גם שם, שאנחנו נקבע כרצוננו, נהוג שמות של תכונות לסמן עם אות קטנה בהתחלה.

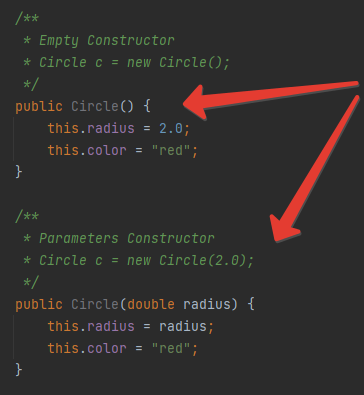
Class Methods



במחלקה ניתן גם להגדיר "שיטות" בהם נשתמש כדי לבצע פעולות במחלקה או עם תכונות המחלקה וערכיה.

בדוגמא לעיל, הוגדרה שיטה לשנות את הצבע של העיגול, השיטה מקבלת פרמטר צבע מסוג מחרוזת ומגדירה את הצבע במחלקה לפיה, השם בפרמטר דומה לשם של התכונה במחלקה, ולכן נצטרך להשתמש במילת המפתח   
this  
כדי להצביע אל התכונה של המחלקה

Class Constructor



מתחת לתכונות יהיו לנו ה"בנאים"

בנאי הוא מתודה המוגדרת עם שם המחלקה ובה משתמשים כדי ליצור "מופעים" של אותה מחלקה,

בנאי יכול להיות ריק, ובנאי יכול לקבל פרמטרים, מחלקה שלא כתובה בה בנאי תקבל בנאי ריק ברירת מחדל בעת הרצת התוכנה.

בנאי ריק: כדי ליצור מופע של אותה מחלקה נוכל להשתמש בבנאי הריק בצורה הבאה:  
Circle c = new Circle();

ניתן לשים לב שבסוגריים אין פרמטרים כלומר השתמשנו בבנאי הריק.

אם נרצה ליצור מופע של אותה מחלקה ולהגדיר בעצמינו גם את ערכי תכונותיה, נוכל להשתמש בבנאים שמקבלים פרמטרים בצורה הבאה:  
Circle c = new Circle(2.0);

בדוגמא לעיל, הבנאי קיבל מספר עשרוני.

**תרגיל:**

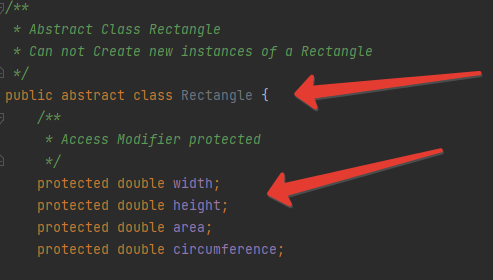
צרו מחלקה   
Person

עם התכונות : גיל,שם,שם משפחה.

צרו בנאי שמקבל את שלושת התכונות בפרמטרים ומגדיר אותם למחלקה.

צרו שיטה ציבורית שמשנה את הגיל במחלקה.

Abstract Class

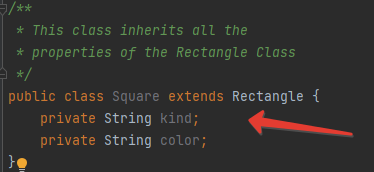


מחלקה אבסטרקטית היא מחלקה שלא ניתן ליצור מופעים שלה, אז למה לנו להשתמש בכזאת מחלקה?

נבחר בעולם הגאומטריה כדוגמא,

בעולם הגאומטריה ישנם המון סוגי מצולעים ולכל מצולע תכונות ומאפיינים משלו, כמו כן ישנם מאפיינים שמשותפים לכל המצולעים לדוגמא שטח והיקף,

נוכל להיעזר בהפשטה וליצור מחלקה אבסטרקטית של מצולע ו"לרשת" ממנה כדי ליצור מחלקות של מצולעים מוגדרים:



בדוגמא לעיל יצרנו מחלקה של מרובע עם תכונות משלו אשר "יורש" ממצולע ובכך יורש גם את כל תכונותיו.

בצורה זו נוכל להגדיר המון סוגי מצולעים בלי להגדיר כל פעם מחדש את התכונות המשותפות לכל המצולעים.

**שאלה:**

נתונה לנו מחלקה אבסטרקטית Person , ומחלקה Employee היורשת ממנה.

איזו מבין שתי המקרים בעייתי?

1. Person p = new Person();
2. Employee e = new Employee();

**תרגיל**:

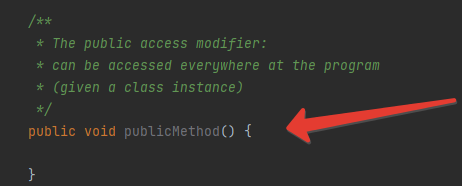
צרו מחלקה אבסטרקטית Animal עם שלוש תכונות שיש לכל בעל חיים,

צרו 2 מחלקות של בעלי חיים שונות עם 2 תכונות ייחודיות לכל אחת שיורשות ממנה.

**Java Access Modifiers**

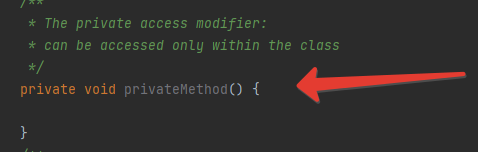
Public

בג'אווה לכל תכונה\שיטה יש משנה גישה שמגדיר למי יש גישה אליה,



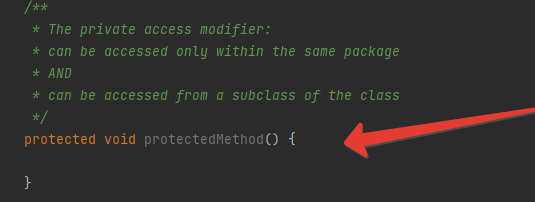
משנה הגישה public משמעותו בשמו, תהיה גישה בכל התוכנה לאותה שיטה.

Private



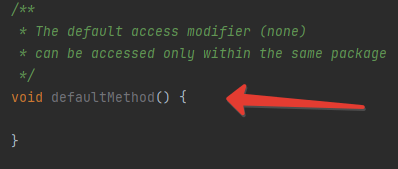
משנה הגישה private גם כן משמעותו בשמו, תהיה גישה רק באותה מחלקה לאותה שיטה\תכונה.

Protected



משנה הגישה protected ייתן גישה למחלקות הנמצאות באותה התיקייה, ולתת מחלקות של אותה מחלקה(מחלקות היורשות ממנה).

Default



בהיעדר משנה גישה, יקרא משנה הגישה default ורק למחלקות הנמצאות באותה תיקיה תהיה גישה.

**The Interface**

****

האינטרפייס, ובעברית – ממשק, הוא סוג מחלקה בג'אווה עם שיטות ללא גוף,