**Debug**

לעתים כשנכתוב קוד וננסה להריץ אותו, התוכנית לא תבצע במדויק את מה שתכננו שהיא תבצע. כמעט תמיד בהרצה ובדיקה ראשונית של התוכנית שנכתוב יהיו תקלות.  
התקלות האלו, נגרמות בגלל טעויות של המתכנת, שכתב את הקוד של התוכנית בצורה לא נכונה. בעיות אלו נקראות **באגים.**

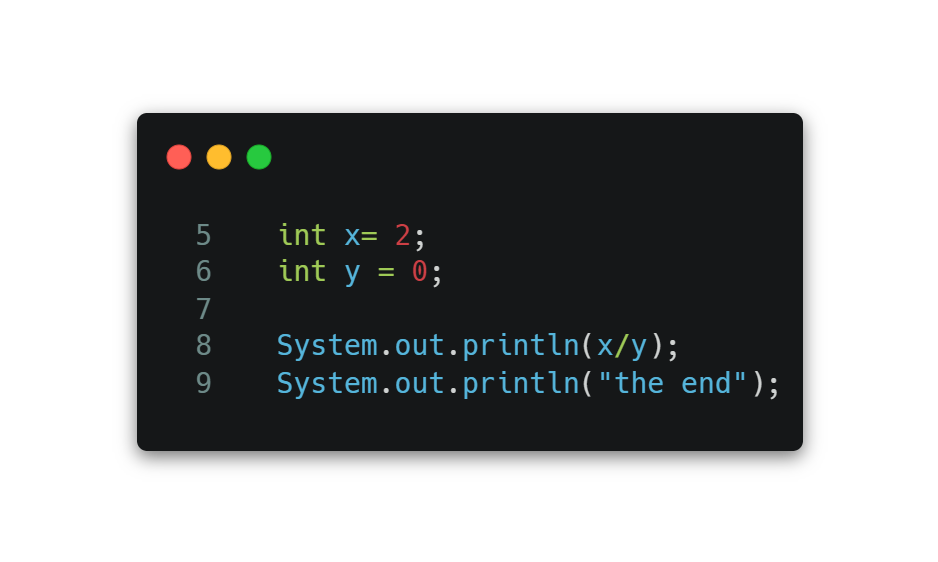
**Debug** מסייע לנו למצוא באגים בתוכנית שלנו בצורה יעילה דרך סביבת העבודה.

יש הרבה שגיאות שניתן לראות בקלות על ידי סימון אדום בקוד. שגיאות אלה נקראות שגיאות קומפילציה ועד שלא נתקן אותן, הקוד לא ירוץ.

לדוגמה, אם ננסה להגדיר משתנה מסוג int (מספר שלם) וננסה לאתחל אותו עם ערך של מחרוזת (String) נקבל את התוצאה הבאה:

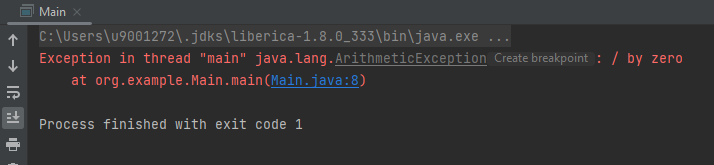
כדי להעלים את הקו האדום ולאפשר את הרצת הקוד, נצטרך לשנות את טיפוס הנתונים של המשתנה.

לעתים, גם עבור שגיאות שמתרחשות **בזמן הריצה** קיים תיעוד המסביר על מיקום הבאג וכיצד ניתן לתקן אותן. לדוגמה:



)ניתן לערוך את הקוד [בלינק הבא](https://carbon.now.sh/?bg=rgba%28171%2C184%2C195%2C0%29&t=seti&wt=none&l=text%2Fx-java&width=469&ds=true&dsyoff=3px&dsblur=9px&wc=true&wa=false&pv=56px&ph=68px&ln=true&fl=5&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code=%2520%2520%2520int%2520x%253D%25202%253B%250A%2520%2520%2520int%2520y%2520%253D%25200%253B%250A%250A%2520%2520%2520System.out.println%28x%252Fy%29%253B%250A%2520%2520%2520System.out.println%28%2522the%2520end%2522%29%253B)(

במקרה הזה, נוכל להריץ את הקוד אבל כשנריץ אותו יודפס הפלט הבא:



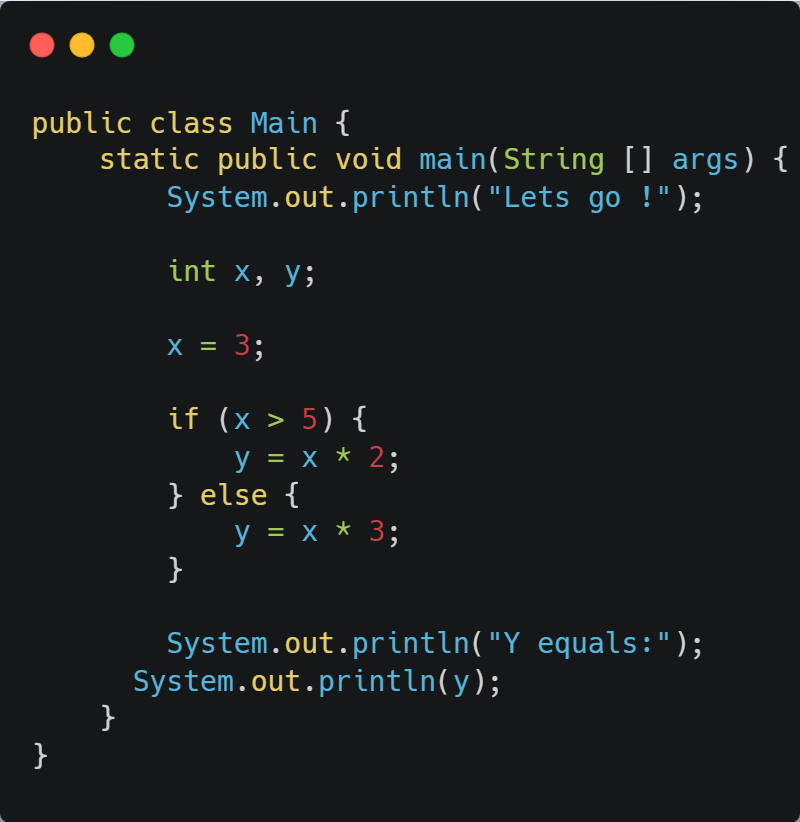
מופיעה **שגיאת זמן ריצה** שאומרת שהייתה חלוקה באפס וזה לא אפשרי. שימו לב שבשגיאה יש גם הפניה לשורה הבעייתית שגרמה לשגיאה (במקרה הזה שורה 8).

לעומת זאת, ישנן שגיאות מסובכות שכמעט בלתי אפשרי לגלות אותן באמצעות הקומפיילר.  
לכן, בעבודה בפרויקטים גדולים המכילים אלגוריתמיקה מורכבת, כדי "לעקוב" אחרי הקוד ולמצוא שגיאות/טעויות או כדי להבין איך הקוד עובד – נשתמש בכלי ה-Debug.

לשמחתנו, אנחנו עובדים בסביבת העבודה intellij אשר מאפשרת יכולות Debug מגוונות עם ממשק משתמש נוח מאוד.

**Debug בסביבת intellij בJava-**

נסו לכתוב קטע קוד קצר, שעושה פעולה כלשהי (באורך של כ-7-10 שורות קוד)

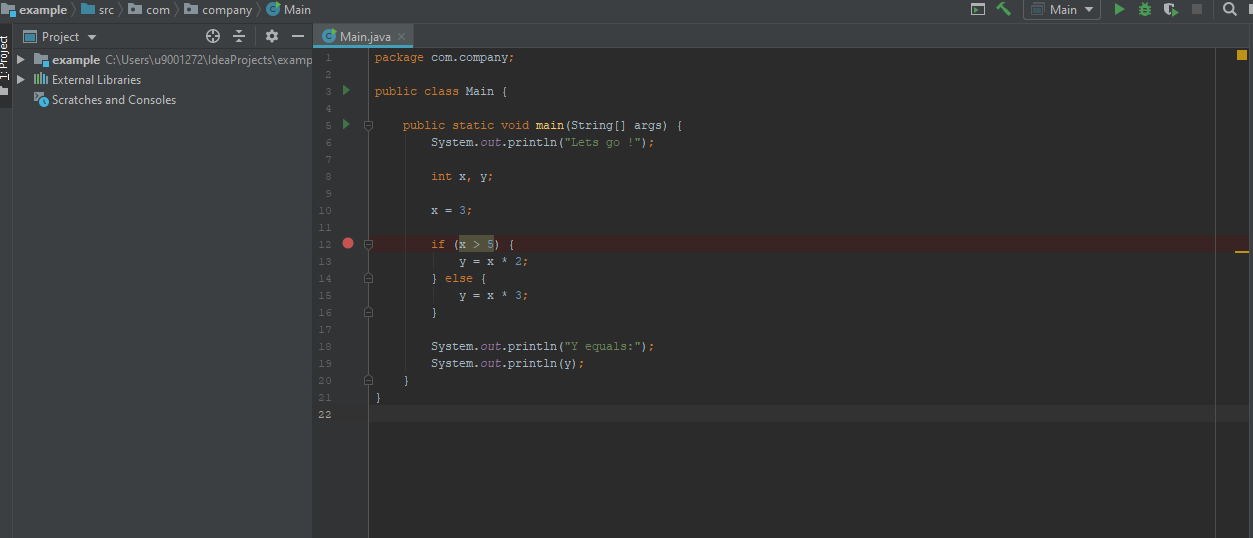
 למשל:

כעת נרצה לדבג את הקוד שלנו.  
כלומר – לעבור אחרי הריצה שלו שורה אחר שורה ולראות את ערכי המשתנים.

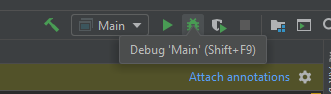
ראשית עלינו להוסיף breakpoint.

**Breakpoint** – נקודה שבה ריצת הקוד תיעצר ונוכל לעקוב אחרי הקוד **בצורה ידנית** מנקודת זו באמצעות debug.

בכדי להוסיף breakpoint , נבחר שורה בקוד שממנה אנחנו רוצים להתחיל לעקוב אחרי ריצת התוכנית, ונלחץ מצד שמאל לשורה – ככה תתווסף נקודה אדומה.  
(הנקודה האדומה היא הbreakpoint)

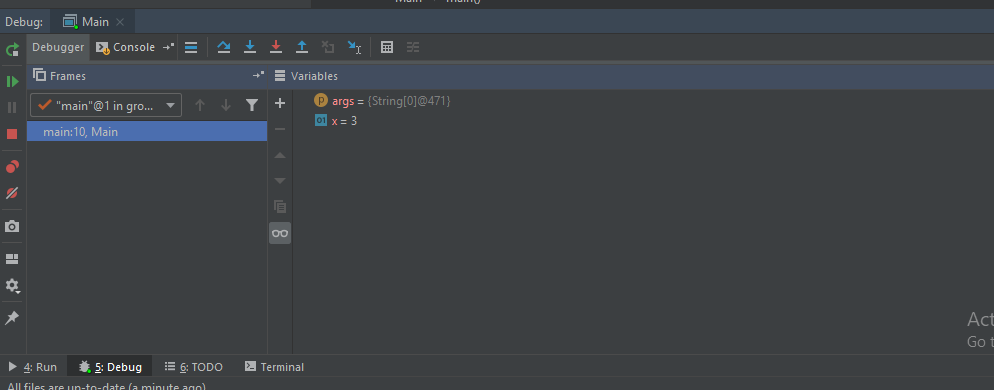


כעת עלינו להריץ את הפרויקט **במצב דיבאג.**  
ע"י לחיצה על הכפתור הבא בסרגל הכלים העליון:



(או ע"י קיצורי המקשים shift + f9)

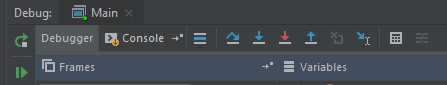
אחרי שהרצנו את הפרויקט במצב debug, נראה את החלון הבא:



זהו חלון הDEBUGGER בו נמצא כל המידע הרלוונטי לתהליך הדיבאג והמצב הנוכחי של התוכנית.

השורה הצבועה כרגע בצבע כחול בקוד שלנו, היא השורה הנוכחית בתוכנית בזמן הריצה הנתון כרגע.

בכדי להמשיך בריצת התוכנית נשתמש בכפתורים הבאים בחלון הDEBUGGER :



הסבר על כל כפתור**:**

* בריבוע האדום- step over (f8): עובר לשורה הבאה בקוד בלי להיכנס לפונקציות בתוך השורה (אם יש כזו). נשתמש בו אם בשורה יש פונקציה מובנית של השפה או פונקציה שאנחנו בטוחים שעובדת ולא רוצים לדבג
* בריבוע הצהוב- step into (f7): עובר לתוך הפונקציה שקוראים לה באותה שורה (אם יש כזו שאנחנו כתבנו והיא אינה של השפה או ספרייה חיצונית).
* בריבוע הירוק- force step into (ALT + SHIFT + f7): נכנס לפונקציה שנקראה מהשורה הנוכחית גם אם היא פונקציה של השפה או של ספריה חיצונית
* בריבוע הכחול- step out (SHIFT + f8): אם נכנסנו לתוך פונקציה ואנחנו רוצים לצאת ממנה ולחזור לשורה שקראה לה נשתמש ב step out.

מוזמנים לחקור עוד באינטרנט על ההבדלים בין הכפתורים.

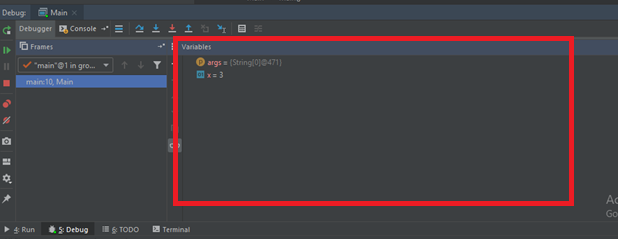
**שימו לב שבמקרה שלנו מכיוון שאין פונקציות שאנחנו קוראים להן, ניתן להשתמש ב step over או step into על מנת להתקדם לשורה הבאה.**

לאחר שעברנו לשורה הבאה, נראה שהפס הכחול מתקדם לשורה הבאה.

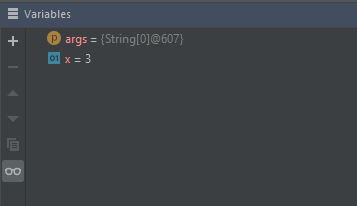
במקרה שלנו התנאי לא מתקיים (אתחלנו את x ב-3 ו3 לא גדול מ-5(, ולכן הדיבאגר מעביר אותנו לelse, כי זו השורה הבאה שתרוץ.

נכיר עכשיו כלים חשובים שיעזרו לנו לראות את הערך הנוכחי של המשתנים כרגע ולדאבג:

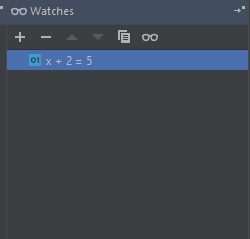
* **חלון הVariables** – מציג את ערך המשתנים שאותחלו ועדיין יש בהם שימוש בתוכנית.



* **חלון הWatchers** – מאפשר לצפות במשתנים שנבחר וגם לעשות עליהם מניפולציות ע"י לחיצה על ה + וכתיבת שם המשתנה בו נרצה לצפות. ניכנס אליו על ידי לחיצה על אייקון המשקפיים:



בתמונה הבאה נגדיר שאנחנו רוצים לצפות בערך של x + 2 ועכשיו לאורך הדיבאג גם אם הערך של x ישתנה גם הערך שיצרנו בעזרת ה watcher ישתנה בהתאם.



יכולת הדיבאג היא מיומנות שדורשת ניסיון והיא חלק אינטגרלי מהעבודה היומיומית של תוכניתן, חשוב להשקיע בה זמן ולתרגל אותה.  
במיוחד בסביבת intellij, קיימות הרבה יכולות debug מתקדמות נוספות וקיצורי מקלדת יעילים שאינם הוסברו במדריך זה.

ישנם המון הסברים ומדריכים באינטרנט על הנושא, תרגישו בנוח לחקור עוד באינטרנט ולגלות בעצמכם.

דוגמה למדריך של debug בintellij [בלינק הבא](https://www.jetbrains.com/help/idea/debugging-your-first-java-application.html).

**בהצלחה!**