**תרגול שבוע 2 – 19.10**

* JDK / SDK
  + JDK = java development kit – ערכה שמכילה את כל מה שאנחנו צריכים כדי לתכנת ולהריץ תוכניות בJAVA
  + מכילה קומפיילר (javac) , VM של JAVA (java), דיבאגר, תיעוד
  + SDK = software development kit
    - JDK היא Java SDK
* JVM – java virtual machine
  + מחשב וירטואלי עם הקצאות זיכרון וירטואליות שאנו מקימים על מערכת ההפעלה שלנו והוא יודע להקצות ולשחרר זיכרון
  + חלק מהJRE
  + JAVA מקמפלת את הקוד שלנו לbytecode שאותו שולחת לJVM שמתרגם אותו
  + לכל מחשב יש JVM משלו, אבל הbytecode הוא אותו דבר עבור כולם
* JRE – Java runtime environment
  + תוכנות וספריות וכל מה שאנו משתמשים בו בזמן ריצה
  + נכלל ב- JDK
* תהליך הקומפילציה ב-java:
  + זמן קומפילציה:
    - הקומפיילר ממיר את הדברים לbytecode
    - הbytecode (קובץ עם סיומת .class) נכנס לJVM שמתרגם אותו לשפת מכונה
  + זמן ריצה:
    - הJVM מעלה את הbytecode שלנו, מכניס אותו לclass loaderף טוען ספריות
    - עובר בbytecode verifier שבודק שהקוד תקין
    - כעת קורים 3 דברים:
      * Interpreter – מתרגם bytecode לקוד מכונה שורה שורה שיורצו מיידית. מתרגם שורה ומריץ אותה (בניגוד לקומפיילר שמתרגם הכל ורק אז מריץ)
      * JIT – just in time compilation – אם הJVM מוצא קטע קוד שחוזר על עצמו, אז הוא מאפשר לעצמו לקמפל בזמן ריצה קטעי קוד שחוזרים על עצמם במקום לעשות כל פעם מחדש שורה שורה
      * Garbage collection
* תהליך הקומפילציה של JAVA יעיל
* JVM – Memory:
  + הJVM מכיל גם זיכרון וירטואלי (מבוסס על הזיכרון של מערכת ההפעלה)
  + יש לנו איזור בזיכרון שמוקצה למידע על מתודות –
    - בשלב של הclass loader הוא מייצר איזור שבו נשמור את המידע על המחלקה
    - גם אם המתודות פרטיות נוכל לקחת את המידע בreflection ולראות איזה מידע כמו איזה משתנים (לא איזה ערכים מכילים) יש.
  + כל האובייקטים נשמרים על הheap (כל מה שנעשה לו new)
  + ב-stack נשמור את כל הפרימיטיבים
  + כל הרפרנסים בJAVA מוקצים בheap וכל השאר מוקצה בstack
  + PC Registers – איזור בזיכרון ששומר מצביעים שמאפשרים ריצה של התוכנית
  + איזור בזיכרון שנועד לnative methods – איזור למקרה שנשתמש במחלקות שנקראו ע"י הJVM אבל מורצות ע"י מתודות של המערכת – נועד לשמירת מידע שחוזר משם
* יתרונות של JAVA:
  + הbytecode תואם לכל JVM
  + אפשר לחזור אחורה מהקמפול ולהבין
* חסרונות של JAVA:
  + שפה יחסית איטית בגלל השימוש בקומפיילר, אינטרפרטר ושימוש בJIT
* Java Garbage Collector:
  + הולך לכל מה ששחררנו ואנו לא משתמשים בו יותר ומשחרר אותם – אוסף את שאריות הזיכרון שהקצנו
  + עובד בצורה אוטומטית – מייצר עץ של פוינטרים וברגע שיש לו קטיעה בעץ הוא הולך ומשחרר את תת העץ שנשאר
  + עושה זאת אחת לכמה זמן ובזמן זה מנתק את הריצה של JAVA
    - גם זה מאט את ריצת JAVA
  + אי אפשר לשלוט בו – אי אפשר לעצור אותו ברגע שהתחיל לרוץ, זו למעשה ישות עצמאית שמתנהלת לבד