**מחלקות ואובייקטים.**

**מנקודת ראות של תוכניתן C, מחלקה class היא מעין הכללה של רשומה (struct).**

**רק שכאן יש דברים נוספים:**

* **"שדה" במחלקה יכול להיות קוד ביצועי (נקרא מתודה)**
* **יש מאפיינים נוספים (בעיקר רמות נגישות).**

**אובייקט - הוא (גולמית לפחות) מופע של מחלקה:**

 **הקצאה ספציפית של זיכרון למימוש הקוד של מחלקה.**

**תכנות OOP מבוסס על כתיבת מחלקות.**

**לכל מחלקה תכנות OOP מבדיל באופן גס בין 2 סוגי קוד:**

* **קוד מימוש של המחלקה**
* **קוד לקוח של המחלקה.**

**לדוגמא, ה-main הוא תמיד לקוח.**

**שדות ה-public של המחלקה נקרא הממשק לקוד הלקוח.**

**קוד הלקוח יכול לגשת רק לשדות שרמת הנגישות שלה היא public.**

**רמות הנגישות העיקריות הם:**

**private**

**protected**

**public**

**רמות הנגישות מצוינות גם לשדות וגם למתודות.**

**התובנה (או המסורת) היא שדות ה-public הם תמיד מתודות.**

**בעצם רמות הנגישות מאפשרות לתוכניתן המימוש לכפות נורמות של הנדסת תוכנה על תוכניתן הלקוח, על ידי כך שנרתום את הקומפילר למטרה הזו.**

**מתודות המחלקה שמוגדרות public נקראות הממשק (interface) של המחלקה.**

**ב-Java מופעים של מחלקות הן תמיד הקצאה דינמית.**

**פרמטרים של מתודות**

**פרמטרים בשפת Java תמיד תמיד עוברים by value כלומר משתכפלים.**

**כאשר מועבר משתנה פרימיטיבי כפרמטר, אי אפשר לשנות את משתנה המקור.**

**כאשר מועבר משתנה מחלקה או מערך כפרמטר, אי אפשר במתודה לשנות משתנה המקור, אבל בהחלט ניתן לשנות את האובייקט עליו הוא מצביע.**

**חבילת Scanner מקביל ל-scanf של שפת C: קריאה מה-standard input.**

**משתנים –**

* **במידה ומוכרזים במתודות – מוקצים (על המחסנית) עם הקריאה למתודה ומשתחררים עם החזרה.**
* **במידה ומוכרזים במחלקה, מוקצים עם ההקצאה ומשתחררים כאשר כל המחלקה משתחררת (כאשר אף משתנה לא מצביע יותר על ההקצאה).**

**ב-java כמעט כל משתנה מופע מאותחל ל"ערך האפס" שלו.**

**למשל ל-int זה 0 שלם, float זה 0.0 ממשי**

**עבור משתנה מחלקה/ מערך/ מחרוזת זה תמיד מצביע האפס null.**

**מקובל בשפה הזאת להעמיד מידע פנימי לרשות קוד הלקוח באמצעות רוטינות get ו-set.**

**זה מאפשר את האופציה של "מידע לקריאה בלבד", "מידע לכתיבה בלבד".**

**סוגי משתנים פרימיטיביים ב-Java:**

* **אין שלמים חסרי סימן.**
* **תנאים הם מסוג מיוחד מסוג boolean.**

**מתודות בונות (נקראות גם בנאים)**

**הן מתודות הנקראות עם ההקצאה הדינמית בתור קוד המאתחל את המופע.**

**הן יכולות לעשות כל קוד שמתודות אחרות יכולות לעשות.**

**מוגדרות עם אותו שם של המחלקה, עם פרמטרים וללא ערך מוחזר.**

**יכולים להיות מספר בנאים עם רשימת סוגי שונים.**

**פקודה נוסך:**

**Account account1 = new Account( 50.00 );**

**גם מגדירה משתנה מחלקה,**

**גם מקצה וגם קוראת לבנאי.**

**פקודה**

**account1 = new Account( 50.00 );**

**רק מקצה וקוראת לבנאי (בתנאי שהגדרתי**

**Account account1;**

**קודם לכן).**