

עקרונות מתקדמים בתכנות

סיכום שיעור

שיעור 3 – תכנות מונחה עצמים – חלק ב'

אל תיקיית השיעור נוסף קוד **ההדגמה** שנעשתה ע"י המדריך/ה. עברו על הקוד, וודאו שהכל מובן ושברור למה עשינו את הדברים בצורה הזו. יכול להיות שתמצאו דרך טובה או אלגנטית יותר לעשות את אותם דברים, נסו אותה!

בנוסף אם יש חלק תיאורטי שמרגיש לכם/ן שלא יושב טוב, אפשר להיכנס ל**מצגת הישנה** (נמצאת בתיקיית השיעור), שם יש עוד חומר תיאורטי שעשוי לעזור לכם. ותזכרו שתמיד אפשר לפנות למדריך/ה.

נושאים שעברנו בשיעור:

- **בנאים ומפרקים:**

- **בנאי** – פונקציה שמאתחלת את שדות האובייקט
 - הבנאי נקרא פעם אחת באופן אוטומטי בעת יצירת האובייקט
 - לא ניתן לעורר את הבנאי באופן ישיר.
- **מפרק** – פונקציה שמשחררת את המשאבים הדינמיים שהוקצו עבור האובייקט
 - המפרק נקרא באופן אוטומטי בעת מחיקת האובייקט:
 - באופן ישיר ע"י הפקודה delete
 - בסיום הפונקציה/חלק הקוד שבו יצרנו את האובייקט (במידה והוא נוצר מקומית)
 - אם אין משאבים דינמיים (לא ביצענו הקצאת זיכרון באמצעות new, פתחנו קבצים או ביקשנו משאבים) אין צורך לממש destructor.

- דריסה והעמסת פונקציות:

- קיום של כמה פונקציות עם אותו שם וחתימה שונה
 - חתימה של פונקציה כוללת
 - שם הפונקציה
 - מסוגי המשתנים והסדר שלהם
 - **לא כוללת את ערך ההחזרה שלה, שמות המשתנים לא משנים.**
- הקומפיילר יודע איזה פונקציה לבחור ע"פ החתימה שלה.

- העתקות:

- **Shallow Copy** (העתקה רדודה) – מעתיקה את שדות האובייקט ביט אחרי ביט, בדומה לפונקציה memcpy שאנו מכירים משפת C.
 - לאחר ביצוע העתקה רדודה האובייקטים זהים בערכי שדותיהם, אבל יכול להיות ששינוי באחד גם ישפיע על השני
- **Deep Copy** (העתקה עמוקה) – מעתיקה את תוכן השדות של האובייקט.
 - לאחר ביצוע Deep Copy אנו יודעים שלא רק שהאובייקטים זהים בשדותיהם, אלא גם לא תלויים אחד בשני (שינוי באחד לא משפיע על האחר).

- BIG 3

- כל מחלקה מגיעה עם 3 דברים "חינם"
 - **בנאי ברירת מחדל** – לא מקבל כלום ולא עושה כלום
 - מאפשר אתחול של האובייקט ללא שליחת פרמטרים
 - לא צריך סוגריים שמשתמשים בבנאי ברירת מחדל
 - ברגע שמממשים בנאי משלנו בנאי ברירת המחדל נדרס (לא ניתן להשתמש בו, אלא רק בבנאי/ם שמימשנו)
 - **בנאי העתקה** – מקבל אובייקט מאותו סוג ומעתיק ממנו את מצבו.
 - כברירת מחדל מבצע shallow copy
 - החתימה של בנאי העתקה היא מדויקת וכוללת רפרנס קבוע.
 - **אופרטור העתקה/השמה** – מעתיק את מצב האובייקט שמצד ימין לאובייקט שמצד שמאל ($a=b$ אומר שהשדות של b מועתקות לתוך a)
 - כברירת מחדל מבצע shallow copy
 - אותה הערה כמו בבנאי העתקה לגבי החתימה.

- דריסת אופרטורים:

- שפת C++ מאפשרת דריסת אופרטורים.
- חשוב להקפיד על חתימה מדויקת
- באופרטורים בינאריים (כאלה שפועלים בין שני אובייקטים) לפעמים צריך להחזיר תוצאת ביניים.
- לדוגמא באופרטור החיבור + מחזירים את `*this` כתוצאת הביניים וזה מאפשר חיבור בשרשרת.