**ב-Java כל הפרמטרים למתודות הם by value**

**ב-C# זה נכון ברמת ברירת המחדל, אבל יש גם אופציות מיוחדות ref, out, params**

**פרמטר ref – המשתנה מועבר לפי יחס (מבחינה טכנית כתובת) ניתן לקריאה וכתיבה**

**פרמטר out - המשתנה מועבר לפי יחס (מבחינה טכנית כתובת) ניתן לכתיבה בלבד. חייבים לכתוב לתוכו ערך במתודה הנקראית.**

**הכרזת פרמטרים עם אופרטורים ref ו-out משנה את החתימה של המתודה.**

**פרמטר params הוא תמיד מערך, מאפשר לי לממש מתודות עם מספר "לא אחיד" של פרמטרים (כמו printf ב-C).**

**ב-C# כמו ב-Java (בדרך כלל) מופעים של משתנים סקלריים ומשתני יחס מוקצים על המחסנית ומידע לא סקלרי על הערימה (בהקצאה דימנית)**

**ס**

**ב-C# יש מערכים (חד מימדיים, רב מימדיים) ואפילו מערך של מערכים (Jagged Array).**

**משתנה מערך הוא הגדרה (די דומה למשתנה מחלקה) בעצם מצביע המחכה לקבל כתובת של הקצאה דינמית.**

**פרמטרים**

**ב-Java פרמטרים מכל הסוגים מועברים תמיד by value (ע"י שכפול).**

**ב-Java לעולם לא ניתן לשנות את משתנה המקור שמועבר כפרמטר.**

**למשל, לא ניתן לממש רוטינת Swap מכל סוג שהוא.**

**ב-C# זה עדיין נכון רק בתור ברירת מחדל: פרמטרים ללא ציון מפורש של מאפיינים מיוחדים מתנהגים בדיוק באותה צורה.**

**המאפיינים המיוחדים נקראים ref, out, params.**

**פרמטר ref פרמטר המועבר by reference כלומר שינוי שלו פירושו שינוי משתנה המקור.**

**במושגים של שפת C זה כאילו בהעברת הפרמטר מוצמד באופן אוטומטי & ומצד שני מוצמד אוטומטית \*.**

**משתנה ref אפשר גם לקרוא וגם לכתוב.**

**למשתנה out אפשר רק לכתוב.**

**params מאפשר לי ב-C# להגדיר מתודות שמקבלות מספר לא קבוע של פרמטרים (כמו printf ב-C).**

**פרמטר params הוא תמיד מערך הנוצר אוטומטית ברגע הקריאה.**

**כל איברי המערך מאותו סוג (לא מגבלה אמיתית כי אפשר להגדיר מערך object).**

**לתוך פרמטר out חייבת להיות כתיבה בתוך המתודה נקראת.**

**מחלקות**

**כמו בשפות האחרות, השדות המוגדרים במחלקה הם מעין משתנים גלובליים עבור המתודות הרגילות (הלא סטטיות או מתודות מופע).**

**הרכבת מחלקות**

**בדרך כלל הבנאי של המחלקה מכילה קוראת לבנאי של המחלקה המוכלת.**

**באופן אוטומטי מוקצה רק משתנה יחס עם ערך null.**

**מאפיין: מתודה שנראית בקריאה כמו שדה מידע מסוג public.**

**מתודת ה-get נקראת כל אימת ש"קוראים" את השדה.**

**מתודת ה-set נקראת כל אימת ש"מציבים" ("=") לשדה, כאשר הערך המוצב נכנס למשתנה מוגדר מראש בשם value.**

**סדרנים indexers**

**אפשר בשפה הזו "להעמיס" אופרטורים, שימוש באופרטור בתוכנית היא קריאה למתודת משתמש.**

**סדרן הוא העמסת ה-[ ].**

**העמסת אופרטורים – קוד שמכיל אופרטור קורא לקוד (מתודה) של המתכנת.**