**תכנות ג'נרי**

**תכנות שמשאיר את סוג הנתון שפועלים עליו פתוח.**

**בכמעט כל שפה פרוצדורלית, אם רוצים לממש אלגוריתם נניח על float ו-double צריך לשכפל את הקוד.**

**במספר שפות (C++, Ada) אפשר לממש קוד ג'נרי בלתי מוגבל באמצעות תמיכה בסוג מיוחד של מקרו.**

**מקרו – מעין קוד שעובר "עיבוד טקסטאלי לפני קומפילציה".**

**ב-Java וב-C# משום מה החליטו לא לממש תכנות ג'נרי מבוסס מקרו.**

**בגרסאות הראשונות לא היה באופן רשמי תכנות ג'נרי בכלל.**

**מאוחר יותר בשני השפות הכלילו תכנות ג'נרי מוגבל.**

**Casting על פרימיטיבים**

**בהקשר הזה casting הוא בעצם קוד ביצועי.**

**Casting בהקשרים אחרים הוא בעיקר התעלמות מסוג הפוינטרים.**

**תכנות ג'נרי ניתן לממש בעזרת המחלקה object (בשפת Java זה Object) שנחשבת לאב קדמון של כל מחלקה (כל משתנה יחס).**

**Boxing ו-Unboxing**

**ב-C# משתנה object יכול להצביע גם על פרמיטיביים (ב-Java אני אזדקק ל-Wrapper).**

**שימוש:**

**Cast\_stack היא מחסנית במימוש רציף של כל סוגי הנתונים האפשריים.**

**הנתונים אפילו לא חייבים להיות הומוגניים.**

**מה אנחנו רוצים שבאמת יהיה אפשר לאכסן בתוצאה של הקימפול רק איברים מסוג מסוים?**

**על מנת ליצור מחסנית שסוג האיברים שלה נשאר פתוח בקוד אבל נקבע ברגע הקומפילציה צריכים את התמיכה בקוד ג'נרי (החל מ-C# 2.0)**