**פולימורפיזם**

* **שימוש חוזר בקוד מבלי לשכפל אותו ולעדכן**
* **עוצמה מרבית לקוד ספריה**
* **גם ניסיון לאפשר למתכנתים לבטא בקוד רעיון מבלי להתחייב בפרטים.**

**מחלקות אבסטרקטיות**

**Abstract classes.**

**לעיתים קרובות מחלקה שהיא קוד משותף למספר מחלקות איננה רחבה דיה להיות מחלקה מלאה.**

**למשל, עובד, עובד אקדמי ועובד מנהלי במכללה.**

**יכול להיות שהמחלקה "עובד" לא מלאה, ולא ניתן להקצות לה מופע.**

**האמצעי מחלקה אבסטרקטית מאפשרת לנו:**

* **להגדיר שדות**
* **לתכנת מתודות**
* **להכריז על מתודות שאין אנו מממשים ו"לקרוא" להם בקוד של מתודות המחלקה.**
* **מחלקה אבסטרקטית: לא ניתן להקצות דינמית, נחשבת למחלקה בירושה וניתן להכריז על משתנה מסוגה שיכולה להצביע על מופעים של מחלקות צאצא שלה.**
* **מחלקה שיורשת ממחלקה אבסטרקטית חייבת לדרוס את המתודות האבסטרקטיות אלא אם כן הן בעצמן אבסטרקטיות.**

**abstract class A**

**{**

 **abstract void am();**

 **void m()**

 **{**

 **am();**

 **}**

**}**

 **class B extends A**

 **{**

 **@override**

 **void am()**

 **{**

 **}// am**

 **}**

**A a = new B();**

**a.m();**