**אמצעי תכנות של Thread-ים**

**אמצעי מניעה הדדית:**

**Lock, Monitor**

**סנכרון מתבצע על מופע ולא על משתנה.**

**ReadWriterLock – מינימזציה של עיקוב של נימים בכך שאפשר לשתף בו זמנית מידע ל-thread-ים שרק קוראים.**

**במידה ומשאב נעול לקריאה בלבד, ניתן שלא לחסום thread –ים שרק רוצים לקרוא.**

**Thread –ים שרק רוצים לקרוא חייבים לחסום רק כאשר thread שרוצה לכתוב למשאב נועל או מנסה לנעול את המשאב.**

**פעולה של thread על עצמו יהיה בדרך כלל קריאה למתודה סטטית של המחלקה Thread.**

**כאשר Thread פועל על Thread אחר, זה יהיה בדרך דרך מתודת מופע.**

**לפעמים אפשר לקבל בפעולות כבדות זמני תגובה טובים יותר ע"י חלוקת העבודה בין thread-ים הודות לליבות המרובות של המחשבים היום.**

**זאת למרות ה-Overhead של ניהול Thread-ים.**

**שיתוף מידע בין thread-ים אפשרי בשתי שיטות:**

* **להגדיר משתנים משותפים כשדות סטטיים של מחלקות.**
* **להקצות מופע של מחלקה שאמורה להיות משותפת ל-thread-ים ולהעביר את כתובת המופע ל-thread-ים.**

**תכנות ג'נרי**

**בעיקרו של דבר תכנות המשאיר את סוג הנתון פתוח.**

**ניסיון לאפשר תכנות על נתונים דומים (אבל שונים) ללא שכפול קוד.**

**הורשה מאפשר לנו לדרוס מתודות אבל סוגי משתנים.**

**ניתן לממש מנגנוני המאפשרים תכנות ג'נרי בלתי מוגבל על בסיס מקרו – שכפול הקוד נעשה אוטומטית ע"י הקומפילר.**

**זה מה שקורה למשל ב-C++ ב-ADA.**

**משום מה זה לא נתמך ככה ב-Java, C#.**