**ממשקים interface**

* **ממשק הוא מאגר של הכרזות על מתודות.**
* **המתודות באופן משתמע הן אבסטרקטיות ופומביות (public) וסטטיות ולא ניתן לדרוס אותם.**
* **מחלקה שמממשת ממשק, מתחייבת לממש את המתודות.**
* **ממשק הוא מעין חוזה שמחלקה מקבלת על עצמה לממש, הוא לא תורם מאומה לממש אותו.**
* **מחלקה יכולה לרשת (להרחיב) רק מחלקה אחת, אבל לא מוגבל של ממשקים.**
* **בשביל מה זה טוב? בעיקרו של דבר, ממשקים נועדו להקל על אינטגרציה של קוד של קבוצות מתכנתים שונות (כמו קובצי .h ב-C):**
1. **מוודאות זהות הגדרות של מתודות משותפות**
2. **ניתן להגדיר משתנים מסוג הממשק: זה מאפשר לקוד מסוים להצביע על קוד חיצוני מבלי לדעת איך הוא ממומש.**

**לדוגמא, לא צריך לדעת את המחלקה המממשת.**

**חריגות Exceptions**

* **בשפה הזו יש מסגרת אחידה לטיפול בתקלות.**
* **חריגות מערכת (כמו חלוקה באפס, גישה דרך פוינטר אפס) נכללת בקטגוריה הזו.**
* **המתכנת יכול "להמציא" סוגי חריגות משלו (בדרך כלל ערכי פרמטרים לא חוקים).**
* **מתודה יכולה לכפות בדיקות תקינות על על הקוד הקורא לה.**
* **אפשר "להתגונן" מפני חריגה, אבל רק ע"י חזרה לקוד שלפני הקריאה לקוד שגרם לחריגה.**
* **ניהול חריגות מאפשר לממש תוכנויות שאינן עפות בשל תקלות, שימוש נכון באמצעי הוא רק לתקלות שלא באשמת התוכנית.**
* **סוגי חריגות הן מחלקות שחייבות להיות צאצא של הממשק Throwable או המחלקה Exception. בדרך כלל יש שרשרת הורשה של מחלקות חריגה שמסווגת את סוגי החריגות ומאפשרת למתכנת לתפור טיפול נפרד לכל סוג.**
* **התגוננות מפני חריגה נעשית בתוך בלוק try … catch. Try מפעיל קוד בצורה שניתן להתגונן, ו-catch היא נקודת קוד ההתאוששות במידה וחריגה קרתה.**
* **יכולים להיות מספר catch-ים המספקים ניהול חריגות לסוגי חריגות שונים בסדר לא יורד של כלליות.**
* **מתכנת יכול לכתוב קוד הגורם לחריגה ע"י האופרטור throw המקבל מופע של חריגה.**
* **מתכנת יכול להגדיר מתודה שבפוטנציה גורמת לחריגה ע"י האופרטור throws. למעט חריגות הנגזרות מ-RuntimeException הדבר כופה על הקורא למתודה לבצע את הקריאה בתוך בלוק try.**
* **בלוקים של try יכולים להיות מקוננים. חריגה יכולה להילכד בבלוק try חיצוני יותר. אם החיפוש של catch מתאים מגיעה ל-main וגם שם היא לא נלכדת, התוכנית עפה.**
* **מילת המפתח throws (להבדיל מ-throw) מופיעה בכותרת של מתודה בכדי להכריז שהמתודה מחזירה סוג מסוים של חריגה.**
* **מילת המפתח throws משמשת ל-2 מטרות:**
1. **לכפות קריאה למתודה בתוך try.**
2. **לפטור מתודה מלממש catch לחריגה מסוימת.**
* **המחלקה המובנית RuntimeException מגדירה חריגות שאין חובה ללכוד ב-try/catch.**
* **למשל, אם מתכנת רוצה להמציא ולזרוק חריגה במתודה *מבלי* לכפות על הקורא למתודה לקרוא רק במסגרת של try/catch אז הוא צריך להגדיר את החריגה כצאצא של RuntimeException.**
* **חריגה שאינה צאצא של RuntimeException נחשבת כ-checked, חייבים להתמודד איתה ואיננה יכולה ישירות להעיף את התוכנית.**
* **חריגות checked בעצם כופים בדיקות תקינות.**
* **מתודה דורסת במסגרת הורשה אינה יכולה להכריז בכותרת על זריקת חריגות שלא היו במתודת האב.**
* **בלוק finally יתבצע בכל מקרה שלא נקרא System.exit(), משמש בעיקר לשחרור משאבים ודאי.**