

Programozás II.

8. labor

Szekeres György

mérnöktanár

Tartalom

- Virtual, override és const
- Ismétlés
- Gyakorlás

Virtual, override és const

- **Override:**
 - Az override kulcsszó a C++11-től kezdődően elérhető, és azt jelzi, hogy egy függvényt felülírunk a leszármazott osztályban.
 - A virtual függvényt felülírhatjuk az override kulcsszó nélkül is, az override biztosítja, hogy a fordító ellenőrizze, valóban egy virtual függvényt próbálunk felülírni. Ha nincs ilyen függvény az alaposztályban, fordítási hiba történik.
- **Előnye:**
 - Biztonság: Az override kulcsszó használatával elkerülhetjük a hibákat, például ha elírjuk a függvény nevét vagy a paraméterek típusát.
 - Olvashatóság: Egyértelművé teszi, hogy a függvényt az alaposztályban definiált virtual függvény felülírására használjuk.

Virtual, override és const

- Virtual:
 - A virtual kulcsszó segítségével jelölhetünk egy függvényt az őssztályban oly módon, hogy az leszármazott osztályokban felülírható legyen.
- Const:
 - A const kulcsszó nem kötelező, de rendkívül hasznos lehet, ha garantálni szeretnénk, hogy a függvény nem módosítja az objektum állapotát. Ez különösen fontos akkor, amikor egy függvény csak adatokat szeretne kiíratni vagy visszaadni, de nem módosít semmit.
 - Mikor érdemes használni?
 - Getter függvényeknél
 - Polimorfizmus esetén
 - Típushibák elkerülése

Könyvtár – feladat

- Készítsen egy programot, amely 3 fájlból beolvassa az adatokat. A fájlokban tároljuk egy könyvtár alkalmazottjait, és olvasóit, valamint a könyveket.
- Hozzon létre egy űsosztályt, amely alkalmas a dolgozók és az olvasók adatainak feldolgozására.
- Származtasson le két gyermekosztályt, amelyek külön kezeli a dolgozókat, és külön az olvasókat.
- Ezen felül készítsen egy Konyv osztályt is, amely a könyveket fogja tartalmazni

Könyvtár – feladat

- Minden osztálynak legyen egy `kiir()` függvénye, amelyet definiáljon felül a leszármaztatásokban
- Írja ki az összes fájl adatát

Megoldás

- Lásd 8.7.cpp

Gyakorlás – időjárás

- Készítsen egy programot, amely egy város napi időjárási adatainak elemzését végzi el objektumorientált módon. Az adatokat egy szöveges fájl, `idojaras.txt`, tartalmazza, amelyben minden sor egy adott nap időjárási paramétereit rögzíti. A fájl adatai szóközzel elválasztva jelennek meg.
- **Idojaras Osztály:**
 - Az osztály feladata az időjárási adatok tárolása és a statisztikák elkészítése.
 - **Adattagok:**
 - `db`: A fájlban található sorok száma (napok száma).
 - `tmb`: Dinamikus tömb az adatok tárolására.

Gyakorlás – időjárás

- A feladatot a következő függvényekkel oldja meg:
 - Konstruktor: beolvassa az adatokat, és létrehozza a dinamikus tömböt
 - Kiír(): kiírja a beolvasott adatokat
 - legeslegNap(): meghatározza, mely napokon volt a legmelegebb időjárás
 - atlagHomerseklet(): Kiszámítja a napi hőmérsékletek átlagát
 - rekordIdo(): kiírja a leghidegebb és legmelegebb hőmérsékleti adatokat
 - Destruktor: felszabadítja a dinamikus memóriát
- Házi feladat!
 - Egészítse ki a legeslegNap() függvényt, hogy a következő adatokat is visszaadja: legcsapadékosabb és legszárazabb nap

Megoldás

- Lásd 8.10.cpp

Köszönöm a figyelmet!