

2. Az objektum elvű programozás definíciója. Absztrakció és specifikációs módszerek. Adattípusok és öröklődés.

Objektum elvű programozás:

Az objektumorientált szemlélet a valóság megközelítésének és ábrázolásának egy módszere. A modellezés során a valós tárgyakat objektumokkal ábrázolja, amelyeket állapotukkal, és metódusaikkal jellemez. Az objektumorientált megközelítés tehát egy programozási szemléletmód, melynek alapján rendszerfejlesztési módszertanokat is kidolgoztak. Ezek a módszertanok a teljes fejlesztési folyamatot átfogják, a megvalósíthatósági elemzéstől kezdve az analízisen, tervezésen és implementáláson keresztül a tesztelés és karbantartás folyamatáig. Az egyes objektum-modellek speciális jellemzőkkel rendelkeznek, amelyek lehetővé teszik, hogy a valós világ egységeihez hasonló módon viselkedjenek. Az objektumorientált szemlélet alkalmazásával, a valós világ és a modell kapcsolatának szorosabbá tételével, nagymértékben megkönnyítjük a valóság megértését, a valósághoz közelebb álló megközelítések alkalmazásával pedig egyszerűbben áttekinthetővé és könnyebben módosíthatóvá válik a program.

Absztrakció

A típusok használatával a programozó lehetőséget kap arra, hogy magasabb szinten gondolkodjon a programról, ne az alacsony szintű megvalósítási kérdések legyenek a meghatározók

Absztrakción a programozás adott szintjén a megoldás szempontjából lényegtelen részletek elhanyagolását értjük. Két fő formáját különböztetjük meg:

1. *Paraméteres absztrakció:* a formális paraméterekhez rendelhető aktuális paraméterek konkrét tulajdonságainak elhanyagolása.
2. *Specifikáció szerinti absztrakció:* az adat konkrét ábrázolásának (reprezentáció) és a rajta értelmezett műveletek konkrét megadásának (implementáció) elhanyagolása.

Specifikáció:

Az a folyamat melyben az objektum leírásához egyedi jellemzőket használunk.

- Szuper- vagy őssosztály
- Al- vagy (le)származtatott osztály.
- Osztályszerkezet létrehozása

Gyakran úgy valósítjuk meg, hogy az egyes alosztályokba további attribútumokat vezetünk be.

Specifikáció öröklés: Az altípusképzés során megengedjük a változtatást is, vagyis azt, hogy az új osztályokban az egyes műveletek átdefiniálhatóak legyenek, implementációjuk, esetleg specifikációjuk is különbözzön az őssosztálybelitől.

Adattípusok:

Az **adattípus** (gyakran röviden **típus**) az értékek egy halmazához rendelt név vagy címke és ezen halmaz értékein végrehajtható néhány művelet.

- elemi adattípusok – a legegyszerűbb típusok
- összetett adattípusok – a típus elemi típusokból épül fel, pl. rekord. Absztrakt adattípusoknak vannak összetett adattípusú és interfész adattípusú jellemzőik is, attól függően, minek tekintjük őket.

Az adattípus fogalma: Az adatabsztrakció első megjelenési formája az adattípus a programozási nyelvekben. Az adattípus maga egy *absztrakt* programozási eszköz, amely mindig más, *konkrét* programozási eszköz egy *komponenseként* jelenik meg. Az adattípusnak *neve* van, ami egy azonosító. Egy adattípust három dolog határoz meg, ezek:

– tartomány – műveletek – reprezentáció

Az adattípusok tartománya azokat az elemeket tartalmazza, amelyeket az adott típusú konkrét programozási eszköz fölvehet értékként. Bizonyos típusok esetén a tartomány elemei jelenhetnek meg a programban literálként. Az adattípushoz hozzátartoznak azok a műveletek, amelyeket a tartomány elemein végre tudunk hajtani.

Az adattípusoknak két nagy csoportjuk van:

A *skalár* vagy *egyszerű* adattípus tartománya atomi értékeket tartalmaz, minden érték egyedi, közvetlenül nyelvi eszközökkel tovább nem bontható. A skalár típusok tartományaiból vett értékek jelenhetnek meg literálként a program szövegében.

A *strukturált* vagy *összetett* adattípusok tartományának elemei maguk is valamilyen típussal rendelkeznek. Az elemek egy-egy értékcsoporthat képviselnek, nem atomiak, az értékcsoporthat elemeihez külön-külön is hozzáférhetünk. Általában valamilyen absztrakt adatszerkezet programnyelvi megfelelői.

Absztrakt adattípus: Az absztrakt adattípus olyan adattípus, amely megvalósítja a *bezárást* vagy *információ rejtést*. Ez azt jelenti, hogy ezen adattípussal nem ismerjük a reprezentációt és a műveletek implementációját. Az adattípus ezeket nem mutatja meg a külvilág számára. Az ilyen típusú programozási eszközök értékeihez csak szabályozott módon, a műveleteinek specifikációi által meghatározott *interfészen* keresztül férhetünk hozzá. Tehát az értékeket véletlenül vagy szándékosan nem ronthatjuk el. Ez nagyon lényeges a *biztonságos programozás* szempontjából. Az absztrakt adattípus (angol rövidítéssel: ADT – Abstract Data Type) az elmúlt évtizedekben a programnyelvek egyik legfontosabb fogalmává vált és alapvetően befolyásolta a nyelvek fejlődését.

Örökítési célt szolgáló, absztrakt metódust tartalmazó osztály. Soha nem példányosítjuk.

Öröklődés:

Minden kategóriának számos alkategóriája lehet, s ezek alatt az alkategóriák alatt is további alkategóriák vannak. Minden szint speciálisabb, mint az őt megelőző, ugyanakkor az is igaz, hogy ha egy tulajdonságot definiáltunk, akkor minden alosztály ez alatt a szint alatt tartalmazza ezt a jellemzőt. Az alosztályok létrehozásánál mindig támaszkodunk annak az osztálynak a tulajdonságaira, amelyiknek az új alosztály része. Az alosztály öröklí az ősoosztály tulajdonságait. Az ősoosztály is lehet, hogy egyúttal egy másik osztály alosztálya, vagyis tulajdonságainak egy részét szintén örökölte. Az alosztályok leírása egyszerűsödik, mivel az örökölt nem kell mindig leírni, egyszerűen újra felhasználjuk a már adott osztályokat. Ekkor az új alosztályt a kiindulásul vett osztály örökösének, illetve a kiindulásul vett osztályt az új osztály szülőjének nevezzük. Az öröklés tehát osztályok közötti alosztályképzési művelet. Az osztályok így öröklődési hierarchiába rendezhetők.

Az öröklődés lehet egyszeres vagy többszörös. Egyszeres öröklődésről pontosan akkor beszélünk, ha a leszármazott osztálynak pontosan egy közvetlen ősoosztálya van. A leszármazott osztály mindig öröklí az ősoosztály adatait és metódusait.