Az OOP(Object Oriented Programming) már az 1980-as években megjelent de még a mai napig nagyon hasznos. Az OOP nélkülözhetetlen eleme lett a szoftverfejlesztésben. Köszönhetően a mindenütt jelenlévő nyelveknek mint java és c++, nem leszel képes szoftvert fejleszteni telefonra ha csak meg nem érted az OOP-t. Persze ugyan ez igaz a komolyabb webfejlesztésre a Python, PHP. Azt gondolhatod magadban miért is van szükséged objektumokra ha tudsz használni egyszerű struktúrákat. Ha ilyen szoftvert írt, akkor a nagy akadályokat részakadályokra bontotta, és külön kódegységekben oldotta meg. Vagy tapasztalata lehet a funkcionális programozással kapcsolatban, amely a kód elemeit pontos matematikai függvényként kezeli, és megakadályozza, hogy más elemeket befolyásoljanak, azaz ne legyenek mellékhatásai. Az OOP, egy teljesen új módszer a problémák megoldására. Az alapkoncepció az, hogy egy program írása helyett létrehoz egy osztályt, amely egyfajta változót és függvényeket tartalmazó sablon. A tárgyak az osztály önálló példányai! Az objektum-orientált programozás előnyei ebben a fajta „kapszulázásban” rejlenek.

Az OOP néhány legfontosabb előnye:

1. Modularitás a könnyebb hibaelhárítás érdekében

Objektumorientált programozási nyelvekkel dolgozva pontosan tudja, hol kell keresni. Ó, az autó tárgya elromlott? A probléma biztosan az autó kategóriában van! ” Nem kell mást végigcsinálni. Ez a kapszulázás szépsége. Az objektumok önállóak, és a funkcionalitás minden bitje megteszi a maga dolgát, miközben a többi bitet magára hagyja. Ez a modularitás lehetővé teszi az informatikai csapat számára, hogy egyszerre több objektumon dolgozzon, minimalizálva annak esélyét, hogy egy személy megismételje valaki más funkcióit.

2. A kód újra felhasználása öröklés útján

Tegyük fel, hogy az autóobjektumon kívül az egyik kollégának szüksége van egy RaceCar objektumra, egy másiknak pedig egy Limousine objektumra. Mindenki külön építi a tárgyait, de felfedezi közöttük a közös vonásokat. Valójában minden tárgy valójában csak egyfajta autó.

Az öröklési technika időt takarít meg:

Hozzon létre egy általános osztályt (Car), majd határozza meg azokat az alosztályokat (RaceCar és Limousine), amelyek öröklik a generikus osztály tulajdonságait.

Természetesen a Limousine-nak és a RaceCar-nak továbbra is megvannak az egyedi tulajdonságaik és funkcióik. Minden osztály külön funkciókat valósíthat meg magának. Mivel azonban mindkét osztály örökli az autóosztály kulcsfontosságú aspektusait, például a „meghajtó” vagy a „fillUpGas” metódust, az öröklődő osztályok egyszerűen újra felhasználhatják a meglévő kódokat ahelyett, hogy ezeket a függvényeket újra megírnák.

Mi van, ha az összes autó objektumot módosítani szeretné, típustól függetlenül? Ez az OOP megközelítés másik előnye. Egyszerűen változtasson az autóosztályon, és minden autóobjektum egyszerűen örököli az új kódot.

3. Rugalmasság a polimorfizmus(átalakulás) révén

Ebben a példában csak néhány funkcióra van szüksége, például a „driveCar”, a driveRaceCar és a „DriveLimousine”. A RaceCarDrivers egyes tulajdonságokat megoszt a LimousineDrivers programmal, de más dolgok, például a RaceHelmets és a BeverageSponsorships, egyedülállóak. Itt játszik szerepet az objektumorientált programozás édes polimorfizmusa. Mivel egyetlen funkció képes alakváltást alakítani, hogy alkalmazkodjon bármelyik osztályhoz, létrehozhat egy funkciót a szülő Car osztályban, „drive” néven - nem „driveCar” vagy „driveRaceCar”, hanem csak „drive”. Ez az egy funkció a RaceCarDriver, a LimousineDriver stb. esetén működne. Valójában még a „raceCar.drive (myRaceCarDriver)” vagy a „limo.drive (myChauffeur)” is működhet.

4. Hatékony problémamegoldás

Egy olyan nyelv, mint a C, elképesztő örökséggel rendelkezik a programozási történelemben. Minél összetettebbé válik, annál nagyobb az esélye, hogy összeomlik. Eközben egy funkcionális stílusú program megírása olyan nyelven, mint a Haskell vagy az ML, nehézkes lehet. Az OOP nyelvek lehetővé teszik a problémák lebontását, amelyeket aztán megoldhat egyesével. Az objektum-orientált programozás nak nagyon sok előnye van. Ha összetett programozási kihívásokat kell megoldania, és kódeszközöket szeretne hozzáadni a készségekhez, az OOP a barátja.