**Object Oriented Programming Nedir?**

Nesne Yönelimli programlama, en temel yapı taşı sınıf nesne ilişkisine dayanan bir programlama şeklidir. Temeli, 1960’lı yıllarda Simula diliyle atılmıştır. Nesne yönelimli programlama, özel yöntemlerle soyut bilgisayar bilimleri kavramaları ile gerçek dünya arasında bağlantı kurarak somutlaştırmasını sağlar. Yani bilgisayarın bir nesneyi insan gibi algılamasını sağlar. Ayrıca somutlaştırma sayesinde karmaşık problemleri insanın anlayabileceği seviyeye indirger. Kapsülleme özelliği ile verilere ve işlevlere doğrudan erişim engelleyerek, yanlış veri ve işlev girişi engeller ve potansiyel veri kaybını önler. Miras alma özelliği ile yeni bir sınıfın eski oluşturulmuş sınıfların özelliklerinden kolayca faydalanılmasını sağlar. Başka bir deyişle sınıflar arası kalıtımı sağlanır. Canlılardaki kalıtım yoluyla genlerini yavrularına aktarılması gibi düşünülebilir. Bir kodun tekrar kullanılabilirliğini arttırır ve başka programcıların kolay adapte olmasına yardımcı olur. Soyutlama özelliği ile bir nesnenin istenen bilgisinin zaman kaybına uğramadan, doğruluk potansiyeli yüksek bir şekilde elde edilmesini sağlar.

**- Nesne Yönelimli Programlama’nın Önemi**

**Geleneksel programlama yöntemleri üretkenliği ve verimliliği azaltıyordu. Ayrıca karmaşık sistemlerinden ötürü öğrenilmesi ve öğretilmesi uzun sürüp zaman kaybına yol açıyordu. Tüm bu olumsuzluklar, geliştirilen programların ve projelerin işleyişinin aksamasına, hatta durmasına neden oluyordu. Gelişen nesnel yönelimli programlama, yazılım geliştirme takımlarının kendi aralarında organize olabilmelerini, geliştirdikleri kodları birtakım araçlarla karmaşadan soyutlayarak birbirilerine ifade edebilmelerine imkân sağlıyor. Ayrıca birçok kişi tarafından kolay anlaşılabilmesi sebebiyle kişi veya gruplara olan bağımlılığın ortadan kalkmasını sağlıyor. Nesne yönelimli programlama, yazılımı daha küçük parçalara bölüyor. Böylece yazılımın gelişim süreci daha kolay takip edilebilir hale geliyor. Oluşturulan sınıflar tekrar tekrar farklı projelerde kullanılarak zaman kaybını önler ve gelişime katkı sağlar. Veriye erişimi sınırlandırarak verinin değiştirilmesini önler ve güvenli bir sınıf oluşturur. İçinde bulunan her nesnenin kendi verileri olduğundan hata bulmayı ve düzeltmeyi kolaylaştırır ve kodu netleştirir.**  **Nesne yönelimli programlamanın olmaması, yazılım geliştirme sürecini daha karmaşık, hatalı ve yönetilmesi zor bir hale getirebilir. Bu nedenle bu programlama şekli, günümüzde birçok alanda yaygın olarak tercih edilen programlama dizileri arasında yer alır. Gömülü sistemler, web frameworkleri, makine öğrenimi ve veri bilimi, veri tabanı yönetimi, kurumsal yazılım sistemleri örneklerden bazılarıdır.**

**-Nesne Yönelimli Programlama’nın Amacı**

**Nesne yönelimli programlamanın temel amacı, istenilen programı küçük parçalara bölerek program geliştiricileri için daha yönetilebilir, tekrar kullanılabilir ve anlaşılabilir hale getirmektir. Programı geliştiren kişilerin veya geliştirilen programa sonradan dâhil olan kişilerin var olan nesne ve sınıfları tekrar oluşturmaktan alıkoymaktır. Bir diğer amacı geliştirilen programın gizliliğini sağlamaktır. Bu sayede programın içyapısını korur. Nesnelerin sadece bazı özellikleri değiştirilebildiğinden tüm sistemi etkilemeden programın geliştirilmesini amaçlar. Yazılıma aktarılacak çözümlerin gerçek dünyada karşılığını bulmasını amaçlar.**

**Threading- Multithreading Nedir?**

**Threading ve multithreading, programların içinde yer alan iş parçacıklarının (threads) eş zamanlı bir biçimde yürütülebilmesidir. Threading programlama sayesinde kodları ardışık çalıştırmak yerine aynı anda işleme koyabiliriz. Bu sayede iş parçacıkları işlemciyi daha iyi kullanıp performansı arttırır. Algoritmaların izole bir şekilde dış etkilerden etkilenmeden çalışabilmesine imkân sağlanır. Daha iyi anlamak için gerçek hayattan örnek olarak bir anneyi düşünelim. Anne bir yandan çocuğuyla ilgilenip bir yandan yemek hazırlarsa kısa zamanda yapması gereken görevleri yerine getirir fakat yemek yapma işini bitirip çocuğuyla ilgilenmeye çalışırsa daha uzun sürede yapması gereken görevleri bitirir. Bu durum verimsizliği görevlerin gecikmesini beraberinde getirir**

**-Threading- Multithreading Neden Kullanılır?**

**Threading- Multithreading disiplini birçok durumda kullanılır. Örneğin yapılan program giriş çıkış işlemlerini bekliyorsa** **(bir dosyadan okuma, ağ istekleri yapma vb. gibi) bu disiplin kullanılır. Threadler birden fazla giriş/çıkış işlemini aynı anda işleyebilir ve programın giriş/çıkış işlemini beklerken diğer iş parçacıklarını yürütmeye devam etmesini sağlar. Threading kullanılmasının burada asıl amacı performansı arttırıp verimlilik sağlamaktır. Bağımsız olarak çalışan birden fazla görev varsa threading disiplini kullanılarak görevleri paralel bir şekilde işleme koyar. Bu özellikleriyle kaynak kullanımını optimize eder, sonuç alma süresini kısaltır, görevlerin düzenli bir şekilde yürütülmesini sağlar. Bu yararlarından ötürü otomasyon ve test araçlarında, veri işlemede, web sunucularında, grafik ara yüzleri gibi birçok alanda kullanılır.**

**Inter Process Communicaton Nedir?**

**Process’ler eşzamanlı, bağımsız veya işbirliği içinde olabilir. Bu process’lerin paylaşılan veya aktarılan verilerin yönetilmesine izin veren yapıya IPC (işlemler arası iletişim) denir. İşletim sistemlerinin farklı işlemlerde birbirleriyle iletişim kurmasına izin verir.** İ**letişimin görevi, veri paylaşımı, senkronizasyon ve iş birliği yapma amacındadır. IPC, bir işlemin başka bir işlemi kontrol edebilmesini sağlarken aynı veriyi tutarsızlık göstermeden kullanımına olanak tanır. Boru hatları, sinyaller, soketler, mesaj kuyrukları IPC yöntemlerinden bazılarıdır. IPC kullanırken veri uyuşmazlıkları, deadlock, eşleme problemleri yaşanabilir ancak birden çok kullanıcının aynı bilgiye erişimini ve kullanıcıların birden çok işlemin yürütmesini sağlar. Bir işi daha küçük parçalara bölerek paralel süreçler yürüterek daha kısa sürede sonuç alınmasına imkân tanır. Bu sağladığı olumlu yanlarından dolayı sosyal medya platformlarında, elektronik pazarlamada, web sunucuları ve istemcileri arasında, işletim sistemlerinde çekirdek ile process’ler arasında ve birçok farklı alanda yaygın bir şekilde kullanılır.**

**Kaynakça:**

**https://academy.patika.dev/courses/oop/oop-nedir**

**https://www.techcareer.net/blog/nesne-yonelimli-programlama-oop-nedir-ozellikleri-**

**https://www.argenova.com.tr/nesne-yonelimli-programlama-oop-nedir**

**https://medium.com/datarunner/pythonda-thread-i%CC%87%C5%9F-par%C3%A7ac%C4%B1%C4%9F%C4%B1-c5364ca6d8d0**

**https://kerteriz.net/python-multithreading-programlama/**

<https://youtu.be/b8BypOBXsvs?si=47FDGLsdQFO_FJm-4>

<https://birolemekli.medium.com/s%C3%BCre%C3%A7ler-aras%C4%B1-i%CC%87leti%C5%9Fim-inter-process-communication-ipc-1686df05baad4>

**https://blog.1845.com.tr/ipc-nedir-ipc-acilimi-nedir-nerelerde-kullanilir/**