Olay güdümlü programlama nedir?



www.yzokul.com.tr

Olay güdümlü programlama nedir?



Olay güdümlü programlama nedir? Programın, bir düğmeye basılması ya da bir mesajın gelmesi gibi olaylara yanıt olarak işlemlerini yürüttüğü bir program yazma biçimidir. Program her zaman tek bir sabit sıra izlemek yerine, bir şey olduğunda tepki verir. Bu kitap fikri sıfırdan, adım adım açıklıyor, böylece ön bilgin olmasa bile anlayabilirsin. Sonunda olayların ne olduğunu, işlemleri nasıl tetiklediklerini ve sıranın neden önemli olduğunu bileceksin.



Bir programı bir dizi işlem olarak düşün. İşlem, programın gerçekleştirebileceği küçük bir eylemdir; örneğin iki sayıyı toplamak, ekranda metin göstermek ya da bir oyunu başlatmak. Bu işlemler, daha sonra bir şey olduğunda çalışacak olan şeylerdir. Olay güdümlü programlamada bu işlemleri yazarsın ve sonra onları olaylara bağlarsın. Program hepsini birden çalıştırmaz; doğru anı bekler.



Olay, gerçekleşen bir şeydir; örneğin bir düğmeye basılması ya da bir mesajın gelmesi. Olaylar, programın dinlediği tetikleyicilerdir. Bir olay gerçekleştiğinde program bir işlemi çalıştırarak yanıt verebilir. Bir olayı sinyal olarak düşün: programa bir şeyin olduğunu söyler. Programın olayın ne zaman gerçekleşeceğini bilmesi gerekmez; yalnızca bekler ve gerçekleştiğinde tepki verir.



Düz çizgili bir program her zaman tek bir düz adım sırası izler. Baştan sona sabit bir sırayla, bir işlemden sonra diğerini çalıştırır. Örneğin bir sayıyı okuyabilir, iki katına çıkarabilir ve sonucu yazdırabilir; her zaman bu sırayla. Bu basit ve öngörülebilirdir, ancak çalışırken olan hiçbir şeye tepki veremez. Olay güdümlü programlama farklı bir yol sunar.



Olay güdümlü bir program her zaman tek bir düz sıra izlemez. Bunun yerine bekleyebilir ve işlemlerin ne zaman çalışacağına olayların karar vermesine izin verebilir. Program farklı olaylar için işlemler kurar, sonra hazır bekler. Bir olay gerçekleştiğinde eşleşen işlem çalışır. Ardından program beklemeye geri döner. Bu, programın zaman içinde birçok farklı şeye, hangi sırayla olursa olsun yanıt vermesini sağlar.



Somut bir örnek düşün: Başlat düğmesi olan bir oyun. Kullanıcı Başlat düğmesine tıklar. Bu tıklama bir olaydır. Tıklama gerçekleştiğinde oyunu başlatan işlev çağrılır. İşlev, olaya yanıt veren işlemdir. Program oyunu kendi başına başlatmadı; tıklamayı bekledi ve sonra harekete geçti.



Tıklama olaydır ve oyunu başlatan işlev onun tetiklediği işlemdir. Onları programa şunu söyleyerek bağlarsın: bu düğmeye tıklandığında bu işlevi çağır. Bu bağlantı, olay güdümlü programlamanın kalbidir. Olay oyun mantığını içermez; yalnızca kullanıcının başlatmak istediğini bildirir. Asıl işi işlev yapar.



Olaylar tetikleyici olarak işlev görür. Her olay, ona yanıt vermesi amaçlanan işlemi çağırabilir. Bir düğme tıklaması için işlem bir oyunu başlatabilir. Bir tuşa basma için bir karakteri hareket ettirebilir. Bir zamanlayıcı için bir saati güncelleyebilir. Hangi işlemin hangi olayla gideceğine sen karar verirsin. Program sonra o olay her gerçekleştiğinde doğru işlemi çalıştırır.



Gelen bir mesaj başka bir tür olaydır. Örneğin bir sohbet programı başka bir kullanıcıdan yeni bir mesaj alabilir. O mesaj geldiğinde program onu ekranda göstererek yanıt verebilir. Programın sürekli kontrol etmesi gerekmez; mesaj olayını bekler. Tıpkı bir düğme tıklaması gibi, gelen mesaj da bir işlemi tetikler.



Arayüzler ve etkileşimli uygulamalar sıklıkla bu yaklaşımı kullanır. Arayüz, bir programın görüp kullandığın parçasıdır; düğmeler, menüler ve pencereler gibi. Etkileşimli uygulamaların yalnızca bir kez tek satırda çalışması değil, kullanıcının yaptıklarına tepki vermesi gerekir. Tıklamaları, yazmayı ve diğer eylemleri beklerler. Olay güdümlü programlama, olayları işlemlere bağlayarak bunu mümkün kılar.



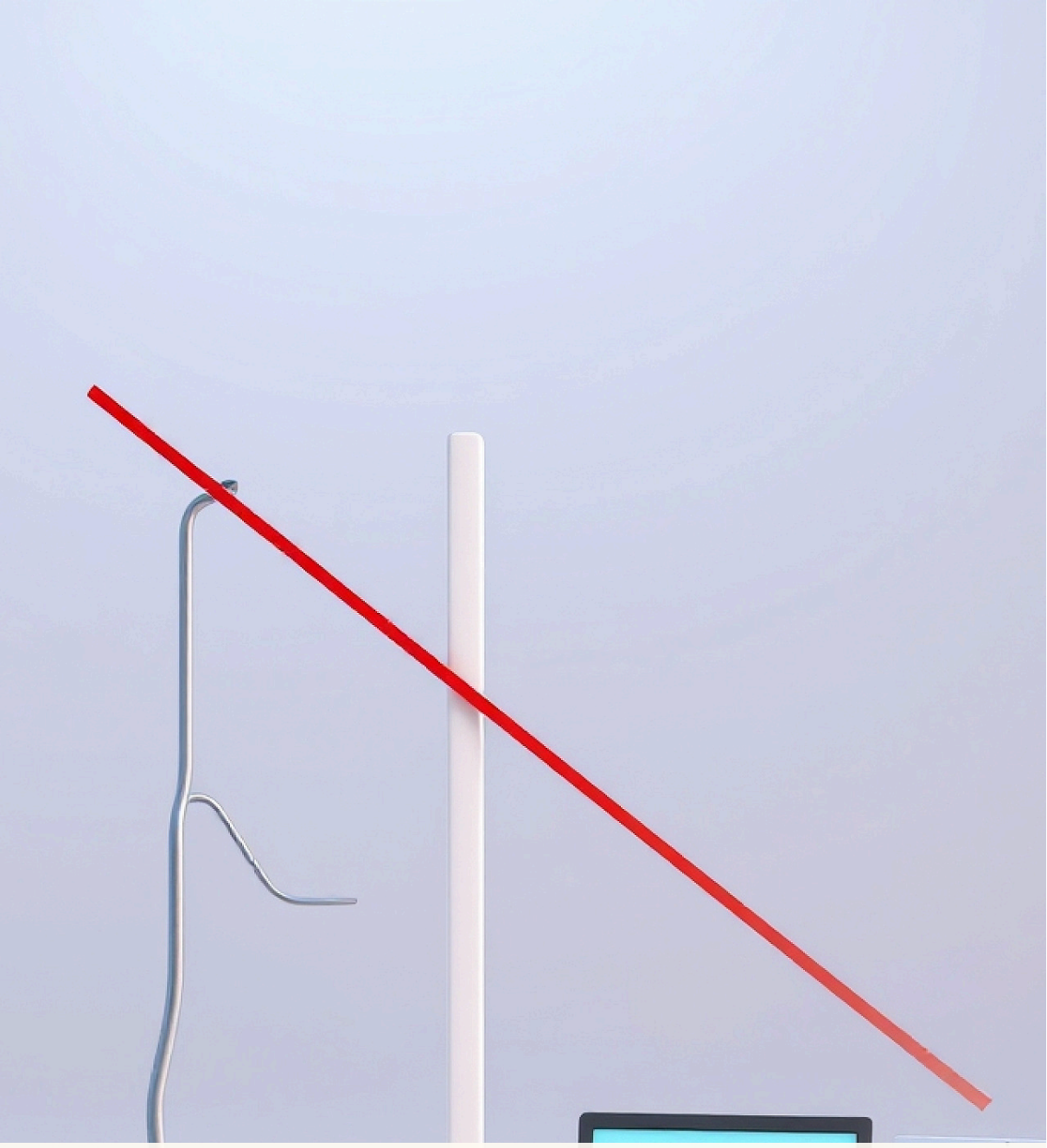
Programın hazır oturup olayları beklediğini düşün. Sabit bir sıradan aceleyle geçmez. Bunun yerine, bir tıklama ya da mesaj gibi bir şey olana kadar boşta kalır. Bir olay gerçekleştiğinde program eşleşen işlemi çalıştırır, sonra beklemeye döner. Bu bekleme durumu normal ve önemlidir. Kullanıcı ya da başka bir sistem bir şey yaptığında programın yanıt vermesini sağlar.



Olaylar farklı sıralarda gelebilir. Örneğin bir kullanıcı bir seviye seçmeden önce ya da sonra Başlat'a tıklayabilir. Programın her yanıtın doğru zamanda gerçekleşmesi için sırayı dikkatle yönetmesi gerekir. Program sabit bir sıra varsayarsa, olaylar farklı geldiğinde başarısız olabilir. İyi olay güdümlü programlar, mantık izin verdiği sürece olayları hangi sırayla gelirlerse gelsinler işler.



Bazen birkaç olay aynı anda ya da birbirine çok yakın gerçekleşebilir. Örneğin bir kullanıcı hızla iki düğmeye tıklayabilir ya da bir tıklama işlenirken bir mesaj gelebilir. Bu eşzamanlı durumların dikkatle yönetilmesi gerekir. Program önce hangi olayı işleyeceğine karar vermeli ve birini işlemenin diğerini bozmamasını sağlamalıdır. Bu, olay güdümlü programlamada temel bir zorluktur.



Yaygın bir hata, programın her zaman tek bir düz sıra izlediğini varsaymaktır. Olay güdümlü programlarda işlemlerin ne zaman çalışacağına olaylar karar verir. Program sabit bir çizgide baştan sona ilerlemez. Bunun yerine bekler ve tepki verir. Bunu unutursan, belirli bir sıra bekleyen ve olaylar farklı geldiğinde bozulan kod yazabilirsin. Her zaman olaylar ve yanıtlar açısından düşün.



Bir başka yaygın hata, olay sırasını ve eşzamanlı durumları göz ardı etmektir. Bunlar dikkatle yönetilmelidir, yoksa yanıtlar yanlış gidebilir. Örneğin iki olay aynı anda aynı veriyi değiştirmeye çalışırsa sonuç kafa karıştırıcı olabilir. Çakışan ya da beklenmedik sıralarda gelen olayları nasıl işleyeceğini planlaman gerekir. Dikkatli yönetim programı doğru ve öngörülebilir tutar.



Özetle: olay güdümlü programlar işlemleri tıklamalar ve mesajlar gibi olaylara yanıt olarak çalıştırır. Tek bir düz sıra izlemek yerine olayları beklerler. Ayrıca olay sırasını ve eşzamanlı durumları dikkatle yönetirler. Başlat düğmesi örneğini hatırla: tıklama, oyunu başlatan işlevi tetikler. Bu anlayışla basit olay güdümlü programları okumaya ve yazmaya başlayabilirsin.