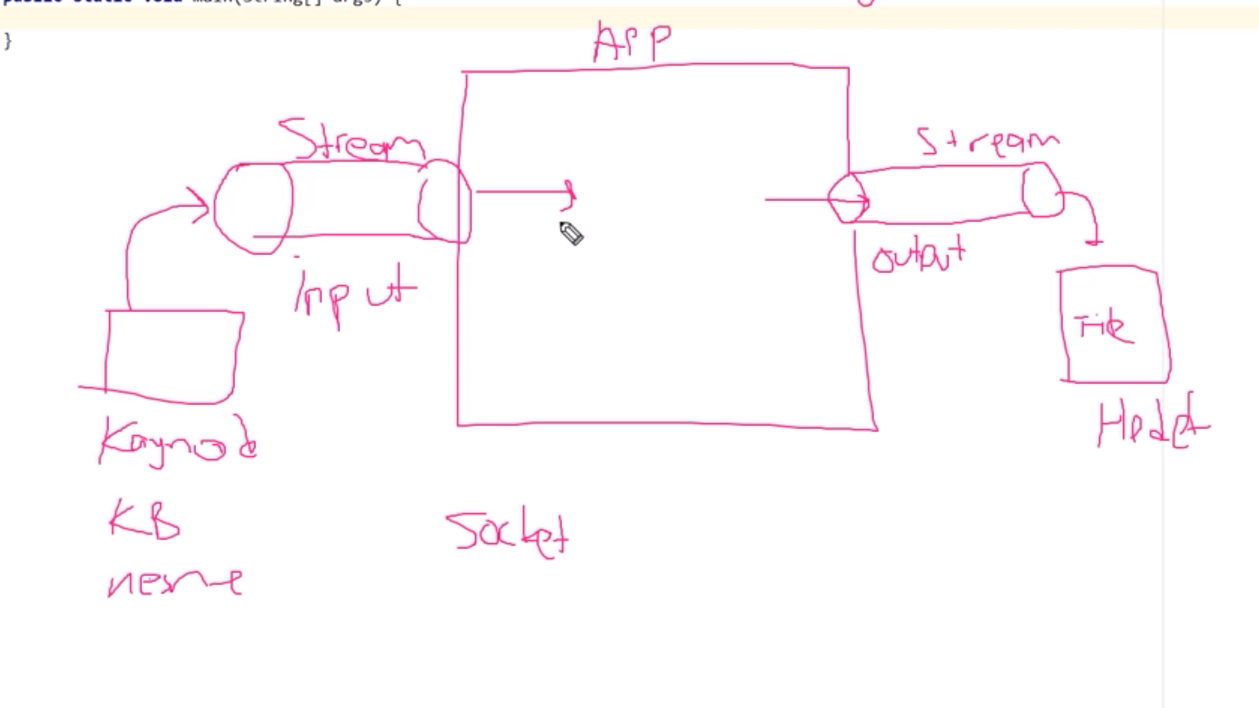
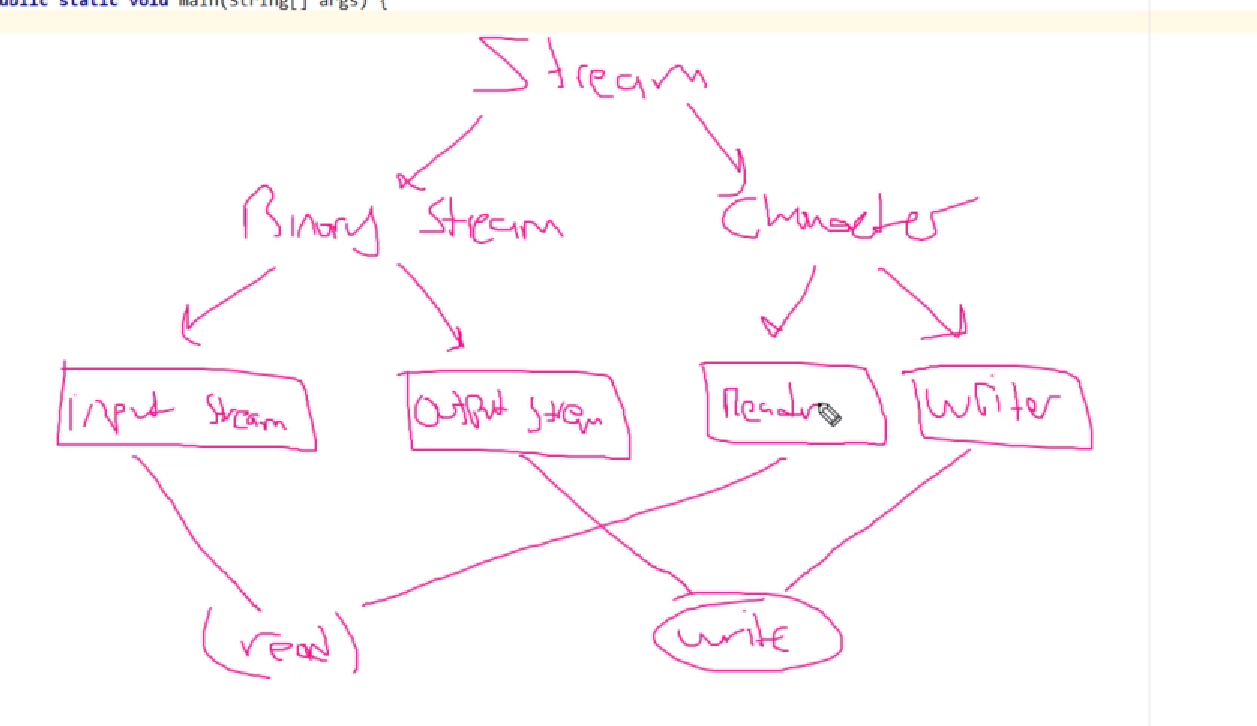
Dosya İşlemleri:

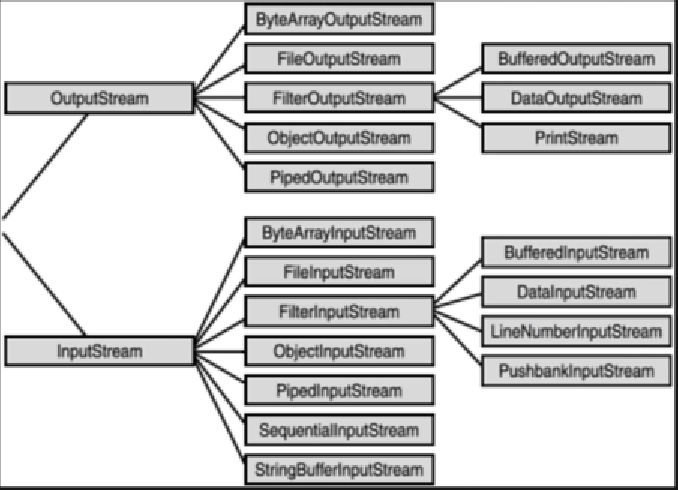
* Şu ana kadar yaptığımız uygulamalarda verilerim RAM’de tutulurdu ve bunlar kalıcı değil uçucuydu.
* Dosyalar, HDD’de tutulur. HDD, Ram’e göre yavaş çalışan ancak kalıcı verilerle çalışmamıza olanak sağlayan bir yapıdır.



* Veriler, dış kaynaktan uygulamaya veya uygulamadan hedefe stream’ler üzerinden aktarılır.
* Stream: Üzerinden verilerin geçtiği mantıksal bir yoldur. Yani veriler stream denen yapılar üzerinden geçerek uygulamamıza girer veya uygulamamızdan çıkar. Streamler, gözle görülebilir bir şey değildir ancak anlamamız için yardımcı olması için streamleri bir boru olarak düşünebiliriz. Ve bu borulardan veriler geçer.
* Uygulamaya dışarıdan veri girişi oluyorsa, input stream’i, uygulumadan dışarıya veri çıkışı oluyorsa output stream’i kullanırız.
* Veriler, streamler üzerinden byte olarak geçiyorsa byte streamsleri, karakter olarak geçiyorsa character streamsleri, object olarak geçiyorsa object streamsleri kullanırız.



* Dosya işlemleri için input-output streamleri kullanırsak veriler dosyalara binary olarak yazılır veya dosyalardan alınır.
* Dosya işlemleri için reader ve writer kullanırsak verilre dosyalara karakter olarak yazılır veya dosyalardan alınır.



* Input ve Output Stream’ler AutoClosable interface’inden gelmektedir. Bu yüzden try with recourses kullanılarak bu tipten dosyaları manuel olarak kapatmamız yerine otomatik olarak kapatılmalarını sağlayabiliriz.
* Dosya işlemlerinde kullanılabileceğimiz class’lar yukarıdaki gibidir. Ancak bunların hepsini öğrenmek gereksizdir. Bu yüzden sadece FileOutput-inputStream, ObjectOutput-Input stream, DataInput-OutputStream ve de verileri okurken veya yazarken buffe’larayarak bu işi yapabilmemiz için BufferedOutput-InputStream’leri öğreneceğiz.

Bufferlama Olayı: HDD’ler çok yavaştır ve dosyalar HDD’lerde saklanır. Biz dosyalaradan verileri okurken veya dosyalara verileri yazarken bunları byte byte yaparsak byte başına HDD ye erişmiş oluruz ki bu uygulamanın yavaş ve kötü performansta çalışmasına sebep olur. Ama biz bu verileri buffer’layarak HDD ye daha az erişmiş oluruz. Bufferlamak, byte’ları bir gumul yapıp biriktirmektir. Yani byte’ları biriktirip biriktirip öyle dosyaya yazıyorum.

* DataOutPutStream, sistem bağımsız olarak her platformda çalışır. (Windows, Mac vb.) O yüzden dataoutputstream kullanalım.