

Fırat Üniversitesi Yazılım Mühendisliği Bölümü Birinci Sınıf Bahar Yarıyılı Sonu Projesi

Öğrenci Adı: Yavuz Samet Kan

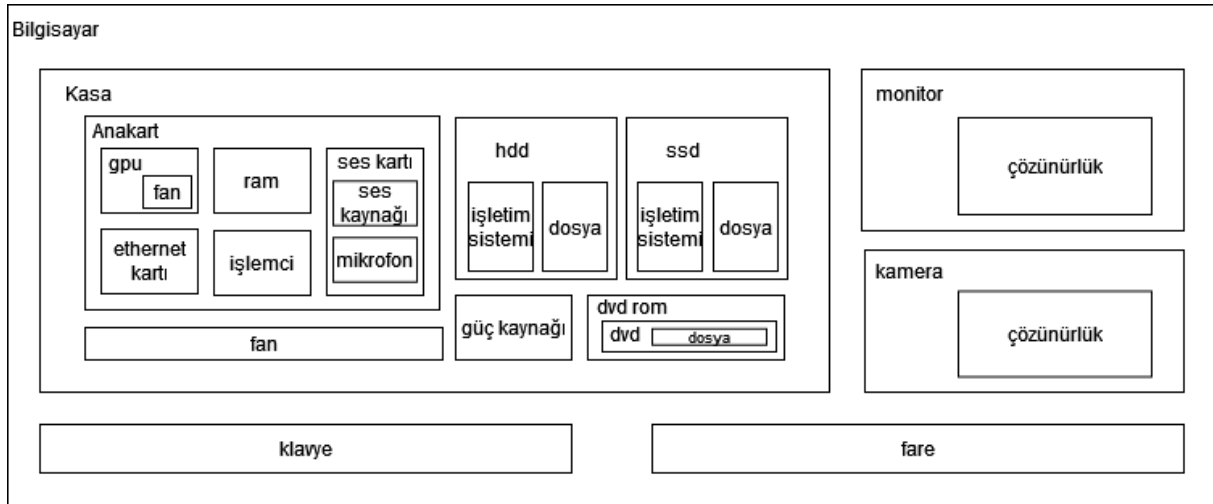
Öğrenci No: 220542011

Bilgisayar Sınıfı ve Özellikleri

Bu proje de bir bilgisayarları OOP (Object Oriented Programming) yaklaşımıyla ele alıp elimden geldiğince özellikler ekleyerek yazılım ortamına aktarmaya çalıştım. Şimdi sizlere bu süreçte izlediğim adımları anlatayım.

1. Adım - Projenin Kompozisyon Taslağı Çalışması

Bu aşamada projenin sahiplik ilişkisini ele aldım ve bir bilgisayarda olabilecek aklıma gelen bütün donanımları eklemeye çalıştım.



İlk Aşamada Projenin Kompozisyonunu Oturtmak için yaptığım Taslak Çalışması

2. Adım – Projenin UML Sınıf Tasarımının Yapılması

Bu aşamada projenin birinci adımında yapmış olduğum fikir edinme kompozisyon taslağını geliştirerek UML tasarımını ortaya koymayı hedefledim.

UML Diagramının SVG hali;

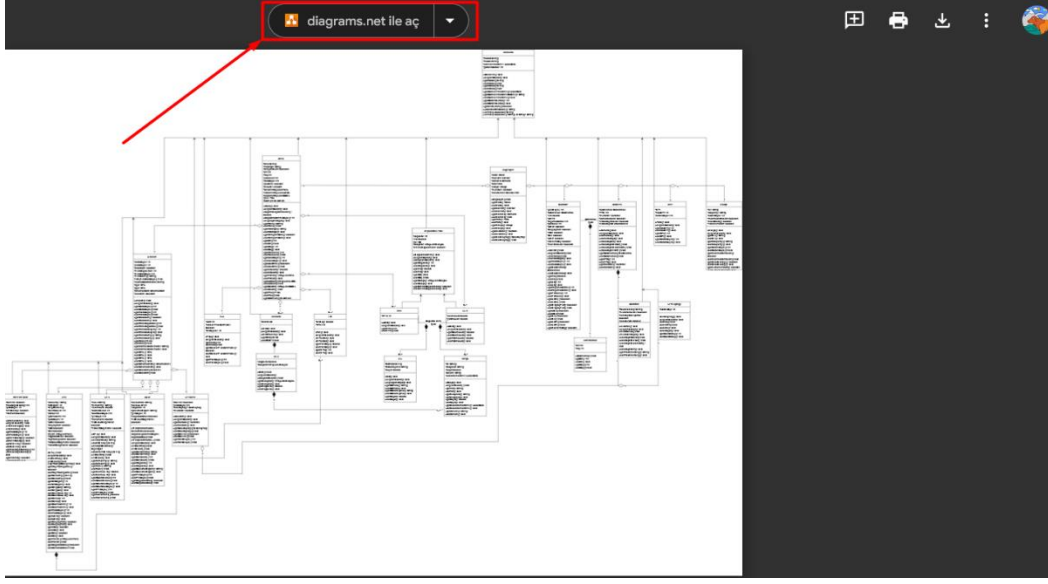
<https://drive.google.com/file/d/1D6XLOZDh5teVnTuwY7j7DD3CT4ow7P6t/view?usp=sharing>

(Bilgisayarınıza indirin ve öyle inceleyin yoksa çalışmaz!)

UML diyagramını detaylı incelemek için yapılması gerekenler;

1.adım -> Link'e tıkla: <https://drive.google.com/file/d/1aqhs0e0Y5v3MawliBxOYF0T-B-U7iHCD/view?usp=sharing>

II.adım -> Gösterilen yere tıklayın ve diagrams programında UML'i inceleyin.



II.adımda anlatılan tıklanması gereken yer

3. Adım – Tasarladığım UML diyagramını programlamak

Yapmış olduğum bu projenin genel amacı dünyada varolan bir objeyi yazılım ortamına nitelikleri vasıtasıyla aktarmak, bu sebeple işlevsel bir program yazmaktan ziyade burada odaklandığım şey düşündüğüm objenin özelliklerini belirlemek ve bu özellikleri efektif bir şekilde yazılım ortamına aktarmaktı. Bu sebeple sınıf tasarımlarını yaptığım objeyi kullanarak bir program geliştirmedim ama bu sınıfı yazılım ortamına aktardım ve bunun kodunu yazdım. Şimdi isterseniz yazdığım sınıfları tanıyalım. (Önemli görmediğim için getter, setter ve constructor metodlarını anlatmadım!)

Donanım Sınıfı – En Genel Süper Sınıf

Baktığınızda bir bilgisayarın bütün birimleri bir donanımdır ve bu birimlerin hepsinin genel ortak özellikleri bulunmaktadır. Bunlar şu şekildedir;

- Marka
- Model
- Satın Alınma Tarihi (bu özellikte LocalDate sınıfı kullanılmıştır.)
- Garanti Süresi
- Garanti Durumu metodu
- Kalan Garanti Süresi metodu

Bu özellikleri efektif bir şekilde programladım ve neredeyse bütün sınıflara OOP'in kalıtım özelliği ile kalıttım.

RAM Sınıfı

Genel olarak benim aklıma gelen bir RAM’de bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır)

- Bellek Türü (Örnek: DDR4)
- Bellek Hızı (Örnek: 3200MHz)
- Kapasite (Örnek: 8GB)
- Gecikme Değeri (Örnek: CL16)
- Pin Sayısı
- Soğutucu Bloğunun olup olmaması durumu (Örnek: true | yani var demek)
- Hız aşırma metodu (örneğin bellek hızını 3200MHz den 3400Mhz’ye aşırarak gibi)
- Hız düşürme metodu (örneğin bellek hızını 3200MHz den 3000Mhz’ye düşürmek gibi)

Bu özellikleri genel istisnalara önem vererek programladım ve işlevsel hale getirdim.

İşlemci (CPU) Sınıfı

Genel olarak benim aklıma gelen bir işlemcide bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır)

- Nesil (Örnek: 12.nesil)
- Mimari Tipi (Örnek: LGA1700)
- İşlemci Hızı (Örnek: 3.7 GHz)
- Cache Bellek (Örnek: 18 MB)
- Çekirdek Sayısı
- Pin Sayısı
- Nanometre (Mimari teknolojisi hakkında bilgi verdiği için önemli bir nitelik)
- Binary To Text metodu (Tanım: 01000001 01010100 01000001 | Değer: ATA)
- Text To Binary metodu (Tanım: ATA | Değer: 01000001 01010100 01000001)
- Binary To Decimal metodu (Tanım: 1000000 | Değer: 64)
- Decimal To Binary metodu (Tanım: 64 | Değer: 1000000)
- Hız Aşırma metodu (örneğin işlemci hızını 3.7GHz’den 3.9GHz’ye aşırarak gibi)
- Hız Düşürme metodu (örneğin işlemci hızını 3.7GHz’den 3.5GHz’ye düşürmek gibi)

Bir işlemci temel bazda 1’ler ve 0’ları yorumlar ve verileri işler bu sebeple işlemci sınıfına binary dönüşüm metodlarını eklemek istedim ve bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Ekran Kartı (GPU) Sınıfı

Genel olarak benim aklıma gelen bir ekran kartında bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır ve ekran kartı, fan sınıfına aggregation ilişkisiyle sahiptir.)

- Bellek Tipi (Örnek: GDDR6)
- Bit Değeri (Örnek: 256 bit)
- Chip Set (Örnek: NVIDIA)
- Çekirdek Hızı (Örnek: 1065MHz)
- Bellek (Örnek: 24 GB)
- Güç Tüketimi (Örnek: 107 Watt)

- Yuva Sayısı (HDMI, VGA gibi slotların totalinin sayısı)
- HDMI çıkışı var mı (Örnek: true | yani var demek)
- DisplayPort çıkışı var mı (Örnek: true | yani var demek)
- DVI çıkışı var mı (Örnek: true | yani var demek)
- VGA çıkışı var mı (Örnek: true | yani var demek)
- Fanlar (Fan sınıfından üretilmiş bir listedir. Fan sınıfı ilerleyen sayfalarda tanıtılacaktır.)
- Soğutucu Bloğunun olup olmaması durumu (Örnek: true | yani var demek)
- Ray Tracing özelliğinin olup olmadığının durumu (Örnek: true | yani var demek)
- Hız Aşırtma metodu (örneğin çekirdek hızını 1065MHz'den 1200MHz'ye aşdırtmak gibi)
- Hız Düşürme metodu (örneğin çekirdek hızını 1065MHz'den 900MHz'ye düşürmek gibi)
- Fanların hepsinin RPM değeri artırma metodu (örneğin RPM değeri 1300'den 1600'e çıkartmak gibi)
- Fanların hepsinin RPM değeri azaltma metodu(örneğin RPM değeri 1300'den 1000'e düşürmek gibi)
- Ray Tracing Özelliğini Açma metodu
- DirectX Açma metodu

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Ethernet Kartı Sınıfı

Genel olarak benim aklıma gelen bir ethernet kartında bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır)

- Anakarta Dahili Mi (Örnek: true | yani dahili demek)
- Maksimum Mbps Desteği
- Yuva Sayısı (RJ45 gibi yuvaların total sayısı)
- Wifi Desteğinin Olup Olmadığı (Örnek: true | yani var demek)
- İnternetin O An Olup Olmadığı (Örnek: true | yani var demek)
- İnterneti Kapatma metodu (Çağırıldığında İnternet Kapanır)
- İnterneti Açma metodu (Çağırıldığında İnternet açılır)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Ses Kaynağı Sınıfı

Genel olarak benim aklıma gelen bir ses kaynağında (hopallör, kulaklık vb.) bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır)

- Ses Düzeyi (En az 0 en çok 100 olabilir | $0 \leq \text{sesDüzeyi} \leq 100$)
- Ses Açma metodu (Çağırıldığında ses seviyesi %100 olur.)
- Ses Arttırma metodu (Her çağırıldığında ses seviyesi %10 artar.)
- Ses Kısma metodu (Her çağırıldığında ses seviyesi %10 azalır.)
- Ses Kapatma metodu (Çağırıldığında ses seviyesi %0 olur.)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Mikrofon Sınıfı

Genel olarak benim aklıma gelen bir mikrofonda bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır)

- Frekan Aralığı (Örnek: 65Hz ~ 18KHz)
- Mikrofon Açma metodu (Çağırıldığında Mikrofon Kapalıysa Açılır)
- Mikrofon Kapatma metodu (Çağırıldığında Mikrofon Açıkça Kapanır)
- Ses Kaydı Başlatma metodu (Çağırıldığında Mikrofon Ses Kaydetmeye Başlar [Amaç görsel bir etki yaratmaktır sesiniz gerçek anlamda kaydedilmez!])
- Ses Kaydı Durdurma metodu (Çağırıldığında Mikrofon Ses Kaydını Durdurur)
- Ses Kaydı Devam Ettirme metodu (Çağırıldığında Mikrofon Ses Kaydını Devam Ettirir)
- Ses Kaydı Bitirme Metodu (Çağırıldığında Mikrofon Ses Kaydetme İşlemini Bitirir)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Ses Kartı Sınıfı

Genel olarak benim aklıma gelen bir ses kartında bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır ve ses kartı; mikrofon, ses kaynağı sınıflarına aggregation ilişkisiyle sahiptir.)

- Anakarta Dahili Mi (Örnek: true | yani dahili demek)
- Yuva Sayısı (Jak girişlerinin sayısı)
- Ses Kaynağı (Ses Kaynağı sınıfından üretilmiş bir objedir): opsiyonel
- Mikrofon (Mikrofon sınıfından üretilmiş bir objedir): opsiyonel

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Anakart Sınıfı

Genel olarak benim aklıma gelen bir anakartta bulunan özellikler ve metodlar bu şekilde; (Donanım sınıfı kalıtılmıştır ve Anakart; RAM, CPU, GPU, Ethernet Kartı, Ses Kartı sınıflarına aggregation ilişkisiyle sahiptir.)

- Sata Sayısı (Bir bilgisayara anakartta bulunan sata sayısı kadar depolama birimi takılabilir bu sebeple sata sayısını kullanıcıdan almak durumundayız)
- Yuva Sayısı (usb vb. yuvaların total sayısı)
- Bluetooth Desteğinin olup olmaması (Örnek: true | yani var demek)
- Maksimum RAM Kapasitesi (Örnek: 64GB | bu bilgisayara totalde en fazla 64GB ram takılabilir anlamına gelir)
- RAM Slot Tipi (Örnek: DDR4 | Takılan RAM'in DDR4 tiptine olması gereklidir!)
- RAM Slot Sayısı (Maksimum kaç adet ram takabileceğimizi bu özellik bize verir)
- RAMler (RAM sınıfından üretilmiş bir dizidir ve uzunluğu RAM Slot Sayısı kadardır.)
- İşlemci Slot Mimarisi (Örnek: LGA1700 | bu mimarideki işlemciler dışında bir işlemci takılamaz)
- CPU (CPU sınıfından üretilmiş bir obje)
- GPU (GPU sınıfından üretilmiş bir obje)
- Ethernet Kartı (Ethernet Kartı sınıfından üretilmiş bir obje)

- Ses Kartı (Ses Kartı sınıfından üretilmiş bir obje)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

İşletim Sistemi (OS) Sınıfı

İşletim sistemi sınıfını tasarlarken aklıma birçok özellik geldi ancak projeyi yaptığım kısıtlı süre, sınav haftasına yaklaşma durumumuz ve maddi olarak bir işte çalışmam gibi sorunlar neticesinde az ama öz olan özellikleri programlayabildim lütfen yaklaşımıma mazur görün.

Genel olarak bir işletim sisteminde bulunan özellikler ve metodlardan birkaçı şu şekildedir;

- OS Türü (Örnek: Microsoft Windows)
- OS Versiyonu (Örnek: Windows 11)
- Boyut (25.5 GB)
- Bilgisayarı Kapatma metodu (Çağırıldığında programdan çıkış yapılır [*System.exit(0)*])

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Dosya Sınıfı

Genel olarak bir dosyada bulunan özellikler ve metodlar bu şekildedir;

- Dosya Türü (Örnek: pdf)
- Dosya Adı (Örnek: rapor)
- Dosya Boyutu (Örnek: 2.3MB)
- Dosya Konumu (Örnek: \Users\Masaüstü)
- Dosya Oluşturulma Tarihi (LocalDate veritipi kullanılmıştır)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Depolama Birimi Sınıfı

Genel olarak bir depolama biriminde (HDD, SSD) bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtlıdır ve Depolama Birimi; OS, Dosya sınıflarına composition ilişkisiyle sahiptir.)

- Kapasite (Örnek: 1024GB)
- İnç (Örnek: 3.5")
- İşletim Sistemi (OS sınıfı kullanılarak oluşturulmuş bir objedir): opsiyonel
- Dosyalar (Dosya sınıfı kullanılarak oluşturulmuş bir ArrayList'tir): opsiyonel
- Birincil Depolama olup olmadığı (Örnek: true | yani birincil depolama alanı | Bilgisayarlarda bir tane birincil depolama alanı olur ve içerisinde işletim sistemi olur bunun belirlenmesi için böyle bir nitelik atadım)
- Kalan Kapasite (Dosyalar ve işletim sisteminin kapladığı boyut sonrasında depolama alanımızda kalan kapasiteyi belirlemek için böyle bir nitelik atadım)
- Dosyaları Gösterme Metodu (Çağırıldığında içerisinde bulunan dosyaları ve dosya boyutlarını sırasıyla gösterir)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

HDD Sınıfı

Genel olarak bir sabit diskte bulunan özellikler ve metodlar şu şekildedir; (Depolama Birimi sınıfı kalıtılmıştır)

- RPM (Örnek: 7200 RPM)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

SSD Sınıfı

Genel olarak bir SSD'de bulunan özellikler ve metodlar şu şekildedir; (Depolama Birimi sınıfı kalıtılmıştır)

- Okuma Hızı (Örnek: 2500MB/s)
- Yazma Hızı (Örnek: 2500MB/s)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Güç Kaynağı (PSU) Sınıfı

Genel olarak bir güç kaynağında bulunan özellikler ve metodlar şu şekildedir; (Donanım sınıfı kalıtılmıştır)

- Watt Değeri (Örnek: 500 Watt)
- Seksen Plus Sertifikasının Olup Olmama Durumu (Örnek: true | yani seksen plus sertifikası var demek)
- Fan Sayısı (Örnek: 1)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

DVDRom Sınıfı

Genel olarak bir DVDRom'da bulunan özellikler ve metodlar şu şekildedir; (Donanım sınıfı kalıtılmıştır ve DVDRom; DVD sınıfına aggregation ilişkisiyle sahiptir.)

- DVD (DVD sınıfının bir objesidir): opsiyonel

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

DVD Sınıfı

Genel olarak bir DVD'de bulunan özellikler ve metodlar şu şekildedir; (DVD; Dosya sınıfına composition ilişkisiyle sahiptir.)

- Kapasite (Örnek: 4.7GB)
- Dosyalar (Dosya sınıfı kullanılarak oluşturulmuş bir ArrayList'tir)

- Kullanılan Kapasite (Dosyaların kapladığı boyutu verir)
- Dosyaları Gösterme Metodu (Çağırıldığında içerisinde bulunan dosyaları ve dosya boyutlarını sırasıyla gösterir)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Fan Sınıfı

Genel olarak bir Fanda bulunan özellikler ve metodlar şu şekildedir; (Donanım sınıfı kalıtılmıştır)

- Fan Çapı (Örnek: 120 mm)
- RPM (Örnek: 1900 RPM)
- RPM Arttırma metodu (örneğin RPM değerini 1900RPM'den 2200RPM'e çıkartmak gibi)
- RPM Azaltma metodu (örneğin RPM değerini 1900RPM'den 1600RPM'e düşürmek gibi)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Kasa Sınıfı

Genel olarak benim aklıma gelen bir kasada bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtılmıştır ve Kasa; Anakart, Fan, HDD, SSD, PSU, DVD Rom sınıflarına aggregation ilişkisiyle sahiptir.)

- Kasa Rengi (Örnek: Siyah)
- Kasa Materyali (Örnek: Alüminyum)
- Kasada Temperli Cam Olup Olmadığı (Örnek: true | yani kasada temperli cam var)
- En (Örnek: 521 mm)
- Boy (Örnek: 503 mm)
- Yükseklik (Örnek: 206 mm)
- Kasada bulunan USB Sayısı
- Kasada bulunan Jak Girişi Sayısı
- Anakart (Anakart sınıfından üretilmiş bir objedir)
- HDD'ler (HDD sınıfından üretilmiş bir objedir): opsiyonel
- SSD'ler (SSD sınıfından üretilmiş bir objedir): opsiyonel
- PSU (PSU sınıfından üretilmiş bir objedir)
- DVD Rom (DVD Rom sınıfından üretilmiş bir objedir): opsiyonel

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Çözünürlük Sınıfı

Genel olarak benim aklıma gelen bir çözünürlükte bulunan özellikler ve metodlar bu şekildedir;

- En (Örnek: 1920 px)
- Boy (Örnek: 1080 px)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Monitor Sınıfı

Genel olarak benim aklıma gelen bir monitörde bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtılmıştır ve Monitor; Çözünürlük sınıfına composition ilişkisiyle sahiptir.)

- Yuva Sayısı (HDMI, VGA gibi slotların totalinin sayısı)
- Çözünürlük (Çözünürlük sınıfından üretilmiş bir objedir)
- İnç (Örnek: 23.8")
- Ekranın Hz değeri (Örnek: 144 Hz)
- Tepkime Süresi (Örnek: 1 ms)
- Parlaklık (300 cd [candela])
- HDMI çıkışı var mı (Örnek: true | yani var demek)
- DisplayPort Çıkışı Var Mı (Örnek: true | yani var demek)
- DVI Çıkışı Var Mı (Örnek: true | yani var demek)
- VGA Çıkışı Var Mı (Örnek: true | yani var demek)
- HDR Özelliği Var Mı (Örnek: true | yani var demek)
- Kavisli Yüzey Var Mı (Örnek: true | yani var demek)
- Monitor Açma metodu (Çağırıldığında monitor kapalıysa açılır)
- Monitor Kapatma metodu (Çağırıldığında monitor açıksa kapanır)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Kamera Sınıfı

Genel olarak benim aklıma gelen bir bilgisayar kamerasında bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtılmıştır ve Kamera; Çözünürlük sınıfına composition ilişkisiyle sahiptir.)

- Çözünürlük (Çözünürlük sınıfından üretilmiş bir objedir)
- FPS (Örnek: 60 FPS)
- Mikrofon (Mikrofon sınıfından üretilmiş bir objedir): opsiyonel
- Kamera Açma metodu (Çağırıldığında kamera kapalıysa açılır)
- Kamera Kapatma metodu (Çağırıldığında kamera açıksa kapanır)
- Video Kaydetme medodu (Çağırıldığında video kaydı başlatır [Amaç görsel bir etki yaratmaktır gerçek anlamda video kaydetmez!])
- Video Kaydı Durdurma metodu (Çağırıldığında devam eden videoyu durdurur)
- Video Kaydı Devam Ettirme metodu (Çağırıldığında durmuş videoyu devam ettirir)
- Video Kaydı Bitirme metodu (Çağırıldığında video kaydını bitirir)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Fare Sınıfı

Genel olarak benim aklıma gelen bir bilgisayar faresinde bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtılmıştır)

- Anlık DPI değeri (Örnek: 1600 DPI | $0 \leq \text{DPI} \leq \text{Ulaşılabilecek Maksimum DPI değeri}$)

- Ulaşılabilecek Maksimum DPI değeri (Örnek: 8500 DPI)
- Buton Sayısı

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Klavye Sınıfı

Genel olarak benim aklıma gelen bir klavyede bulunan özellikler ve metodlar bu şekildedir; (Donanım sınıfı kalıtılmıştır)

- Klavye Dili (Örnek: Türkçe Q)
- Klavye Tipi (Örnek: Mekanik)
- Tuş Sayısı
- Multimedia Tuşları Var Mı (Örnek: true | yani var demek)
- Bilek Desteği Var Mı (Örnek: true | yani var demek)
- Numerik Tuşlar Var Mı (Örnek: true | yani var demek)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Bilgisayar Sınıfı

Bu sınıf tüm sınıfların süper sınıfıdır içerisinde tüm sınıfları hem dolaylı yoldan hem direkt barındırır ve bu sınıf üzerinden tüm bilgisayar özellikleri yönetilebilir. Bu sınıfın özellikleri aşağıdaki gibidir;

- Kasa (Kasa sınıfından üretilmiş bir objedir)
- Monitor (Monitor sınıfından üretilmiş bir objedir)
- Kamera (Kamera sınıfından üretilmiş bir objedir)
- Fare (Fare sınıfından üretilmiş bir objedir)
- Klavye (Klavye sınıfından üretilmiş bir objedir)

bu özellikleri genel istisnalara önem vererek programlayıp, işlevsel hale getirdim.

Son Söz

En nihayetinde 25 sınıflı bu projeyi geliştirdim ve teslim ettikten sonra üzerine çeşitli eklemeler yapmayı da planlıyorum, bu eklemeleri takip etmek isterseniz github linki aşağıdadır;

...

Güzel bir projeydi bana OOP bakış açısı kattığına inanıyorum ve kendimi OOP konusunda geliştirmeye devam edeceğim. Bu güzel proje ödevi için değerli bölüm hocalarıma teşekkür ediyorum, umarım siz değerli hocalarım da geliştirdiğim projeden memnun kalmışsınızdır. Sağlıcakla kalın.