

SUT25 Teorihandboken: Izabella Kondili-Sturesson

1. Vad är .NET och hur länge har det funnits?

.NET är en öppen källkodsutvecklingsplattform. Man kan koda i C#, Python med flera. .NET används för att bygga webbapplikationer likasom mobil applikationer och skrivbords appar.

2. Vad är skillnaden på .NET och C#?

Skillnaden på .NET och C# är att .NET är plattformen man använder för att skriva programmerings språket C#.

*Här är det intressant för mig då jag försöker 'break it down' genom att säga att jag använder VS programmet för att skriva C# kod i en '.NET värld' så att säga.

3. Vilka möjligheter/fördelar finns det med att utveckla i .NET?

Det visar sig vara mångfaldiga. Nu är jag grön i detta och lär mig HEELA tiden om denna mystiska värld.

.NET tillåter programmerare att bygga program så de fungerar på olika operativesystem: Windows, Mac eller mobil.

Det funkar att välja mellan t ex C#, F# eller VB

.NET har färdig kod, vilket innebär att vi programmerare kan ta ifrån detta biblioteket istället för att programmera just en del av programmet på nytt igen.

Det är inte bara appar man kan bygga i .NET, utan man kan bygga små spel såsom våra två första labbar. Vi har även möjlighet att bygga appar på nätet eller jättestora system som företag använder.

I om med det redan finns 'färdiga bitar' i biblioteket, så går det snabbare att bygga ett program, vilket gör det roligare. Det är bara att man sakta men säkert ska lära sig navigera och förstå med.

Till sist, eftersom det är många som använder .NET, anses en bli mera värdefull om man kan bygga med den, ergo, har man lättare att få jobb. (håller tummar och tår).

4. Det finns flera olika varianter/versioner av .NET. Förklara vilka dessa är och hur de skiljer sig åt.

Idag fick vi lära oss av Reidar att det finns fyra (med utrymmer för .NET(5+)) .NET versioner: .NET Framework, .Net Core, .NET(5+), Mono/Xamarin.

Vi börjar med farbror .NET Framework som va Windows—bunden. Farbror har utvecklats till en världsresande kusin (Core), till en modern arvtagare nya .NET 5+.

.NETs utveckling handlar om att ha samlat släkten (.NET Framework, Core, .NET 5+) för att nu kunna bygga appar för nästan allt: dator, webben, mobilen liksom molnet.

5. Vilka varianter/versioner av .NET är aktuella i dagsläget?

Jag kunde avslöja fyra: .NET 8, .NET 9, .NET 10, .NET Framework 4.8

6. Lista de datatyper och datastrukturer som du känner till i .NET/C# och förklara var och en av dessa lite kort.

Int: siffror/heltal

Double: decimaler

String: detta är text även inclusive siffror, som datorn uppfattar som två separata karaktärer så att säga.

Bool: sant eller falskt / på och av / ja eller nej / upp eller ner. Det är en funktion vill jag säga precis som en lampa. Beroende på vad programmet ska göra, anpassar man bool till dess syfte.

*Datastrukturer: här blankade jag först. Så positivt trött i huvudet. Vi har inte gått igenom datastrukturer så mycket SUT25 just ännu, dock känner jag igen ett par stycken från Programmering 1 my Python kursen jag tog för ca ett år sedan:

1) Array: en datastruktur som låter mig lagra flera värden av samma typ i en enda behållare.

Ex: vi har en stående toalettrulle behållare. I den får man bara plats med 5 hela rullar. Eller så kan man ha den tom, eller bara ha 3 toarullar i den. Men toalettrulle behållaren har en fast storlek så att säga.

Jag tänker att en Array kan man använda i ett program som ska lagra hur många bussstolar är upptagna (dvs hur många resenärer en turistbuss har fått beställning på). En av turistbussarna vi har på mitt förra jobb har bara 62 stolar. Då skapar man en array med max antal stolar till 62.

2) List: är en dynamisk samling som kan växa eller krympa.

När jag läser Lista i C#, då det kan krympa eller växa, tänker jag på pengar man har på banken. Lönen är tacksam, då växer listan. Man betalar räkningar, och det krymper igen. Detta är applicerar systemutvecklare när de skapar en hemsida t ex.

7. Vad är skillnaden mellan en kodeditor och en IDE?

Så som jag förstår det finns kodeditor i .NET Visual studio som ett hjälp medel med specifikt koden. När koden blir rött undertill, kan man 'fråga kodeditor' som visar åt sidan vad 'datorn' inte förstår och då får jag 2 eller fler alternativ på hur jag kan

redigera koden så det funkar bättre (för att hitta rätt kod till programmet jag vill uppnå).

IDE betyder Integrated Development Environment.

Det innebär att man använder sig av en lättare version av till exempel visual studio eller VSCode.

IDE har en keditor och andra verktyg som en kompilator, en debugger och testverktyg. Jag använde en IDE under online kursen jag tog Programmering 1 med Python.

Så, IDE, är ren lätthanterlig, omfattande och integrerad lösning för att hantera hela utvecklingscykeln. Keditor fokuserar på att skriva och redigera själva koden.

VG -FRÅGA

Allting handlar egentligen om utveckling i vår teknologiska värld.

Generellt sätt så har vi människor utvecklats framåt ända sedan the Big Bang. Så varför skulle inte teknologin göra detsamma? Människans behov utvecklas, och det innebär att utvecklare-domänen behöver anpassa sig för att hitta relevanta lösningar.

.NET introducerades till världen av Microsoft i början av 2000-talet som .NET Framework. Vi har lärt oss att detta är ett kraftfullt ramverk som även är Windows-bundet. Det betyder att utvecklare kunde programmera i C# och F# bland annat men enbart på Windows datorer. Alltså inte Mac och Linux.

Antal utvecklare som började arbeta i molnet växte. I takt med detta märktes ett behov av att arbeta plattformsoberoende. Då skapades .NET Core år 2016. .NET Core versionen var snabbare, den var öppen till alla som ville bland annat använda/dela och modifiera (man behövde köpa license till detta). Då kunde man plötsligt använda .NET på flera operativsystem: alltså Mac, Windows, Linux. Så detta är ett stort steg för Microsoft. Detta blev ju banbrytande.

År 2020 skedde en shift.

Microsoft samlade alla sina versioner .NET Framework + .NET Core för att göra .NET (aka .NET5..+). En gemensam plattform.

.NET är idag ett generellt namn på ramverket oberoende på de versioner som kom: .NET 5, 6, 7 och 8.

Olika Perspektiv:

Många företag har gamla system byggda i .NET Framework. Det vi förstår genom tidsaspekten är att det tar långt tid att uppgradera från .NET Framework till de nya .NET-versionerna (speciellt om systemet är stort). Vill man lyssna till utvecklarens perspektiv, så villhöver vi nya versioner (som .NET 8) då det är snabbare och enklare att arbeta med. Vi har även kompatibilitets frågan. Nu är jag nybörjare i denna domän, och det jag förstår är att vissa Kod fungerar inte ännu i de senaste .NET versionerna.

Så, många arbetar fortfarande i det gamla framework då det fungerar fortfarande att skapa program. Men de senaste versionerna är mer önskevärda för utvecklare att arbeta med pga bland annat prestanda och att de fungerar på diverse operativsystem: Windows, Mac, Linux, Moln, Web, appar.

Det är dyrt och riskfyllt att bygga om allt från grunden. Därför fortsätter utvecklare använda äldre versioner. Men det mest optimala för SUT25, är att vi arbetar med våra project i .NET 8 (Long Term Support).

Utveckling är oundvikligt. Det Kanske blir ett gang utvecklare som hittar ännu en game-changing sätt att uppgradera de flesta system och underlätta framtida arbete. 😊

*** Jag försökte länka uppgiften via <http://truedocs> grejen...men den tog lika lång tid att ladda som den första dagen vi alla försökte oss på den på skolan.

Därför skickar jag in mina uppgifter istället via Learnpointi gen. Hoppas du har overseende med detta.***