**IT FUNDAMENTALS – PROGRAMMING BASICS**

1- Computational Thinking

Hesaplamalı düşünme, karmaşık problemleri ve problemlerin çözümlerini algoritmik ve sistematik olarak ifade ederek problem çözme sürecinin sağlıklı şekilde tamamlanmasını sağlayan bir yöntemdir.

21.yüzyıl yetkinliklerinin önde gelenlerinden biri olarak kabul edilir. Yalnızca bilgisayar ortamında karşılaşılaşan sorunlarda değil, günlük yaşantıda karşılaşılan her türlü problemi çözerken kullanılabilir. ABD'li bilgisayar bilimci Jeannette Wing'in, hesaplamalı düşünmenin temel bir beceri olarak her öğrenciye öğretilmesi gerektiği konusundaki iddiasını 2006 yılında öne sürmesinden sonra bir çok ülkede ilk ve orta öğretim programlarına hesaplamalı düşünme öğretimi yerleştirilmiştir

Hesaplamalı düşünme genel hatlarıyla şu şekilde gerçekleşir: problem parçalara ayrıştırılır, ayrıştırılan parçalar daha önce karşılaşılan problemlere benzer şekilde çözülebilsin diye karşı karşıya getirilir, ortaya çıkan problemle ilgili olmayan bilgilerin işin içerisinden çıkarıldıktan sonra sorunu çözmek için adımlar oluşturulur ve ortaya çıkan çözüm yolu en verimli şekilde kullanılır. Her karmaşık problem aynı yolla çözülemeyeceğinden hesaplamalı düşünme uygulanırken izlenecek yol probleme göre değişikliğe uğrayabilir.

Hesaplamalı düşünme Ayrıştırma, Örüntü Tanıma, Soyutlama ve Algoritma tasarımı olmak üzere dört bileşene sahiptir.

**Decomposition**

**Ayrıştırma**

Karşımıza çıkan karmaşık bir problemi parçalarına ayırarak daha yönetilebilir hale getirilen aşamadır. Kompleks problemler, büyük sorunlar ile karşılaşıldığında ayrıştırma yapmak vazgeçilmez olmalıdır. Parçadan bütüne gitme mantığı ile ayrıştırılan parçalar diğer aşamalara göre sıralanır ve çözüme ulaşmak için izlenen yol daha kolay ve kısa bir hâle gelir.

**Pattern Recognation**

**Örüntü tanıma**

Terim anlamı olarak bir serideki verilerin aralarındaki örüntüyü, kuralı tanımlama olarak ifade edilebilir. Hesaplamalı düşünmede ise problem içerisindeki benzerlikler tanımlanır ve daha önce karşılaşılmış olan veya elde bulunan problemin alt parçalarında belirlenmiş olan bir problem ile benzerliğinin olup olmadığı bu aşamada belirlenir. Eğer bir benzerlik tespit edilirse önceden belirlenen çözüm tekniklerini bu problemde de kullanmak çözüme yaklaşmada işimizi kolaylaştıracak ve bize zaman kazandıracaktır. Bu bileşen karşılaşılan sorunlara etkili çözümler üretebilmek için önemli bir basamaktır

**Abstraction and Pattern Generalization**

**Soyutlama**

Soyutlama ile çözümün en temelinde olan ve en ihtiyaç olan bileşenlerine odaklanılır. Ayrıntılara göz ardı ederek amacıyla bizi çözüme ulaştırmayacak her şeyi ayırarak sadece önemli öğelere dikkat çekilir. Bu basamak sayesinde odak noktamız sadece belirlenen sorunun çözüm sürecine çekilir ve problem çözme sürecinin detaylardan ayrılarak sade bir hâle getirerek hızlanmasını sağlar.

**Algorithm Design**

**Algoritma tasarımı**

Hesaplamalı düşünmenin son bileşenidir. Bu basamakta belirlenen problemin adım adım nasıl çözüme ulaşacağı belirlenir. Bizi çözüme ulaştıracak olan algoritmayı tasarlarken en kısa yol ile gerçekleşecek olan çözümün belirlenmesine dikkat edilmelidir. Algoritma sadece bilgisayar ortamında kullanılan bir kavram olmamakla beraber hayatımızın her alanında bu bileşene yer verdiğimiz bilinmektedir. Günlük yaşamımızda bir işi gerçekleştirirken sırayla yapmamız gereken adımların bulunduğu her olayda algoritmadan söz edilebilir.

Algoritma tasarımı basamağının içerisinde ele alabileceğimiz başka bir bileşen olarak da hata ayıklamadan bahsedebiliriz. Hata ayıklama tasarladığımız algoritmayı deneyerek belirlediğimiz hataları düzeltme yoluyla algoritmamızı iyileştirmeye yarar sağlar.

2- Algoritma

Algoritma, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol. Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir. Genellikle bilgisayar programlamada kullanılır ve tüm programlama dillerinin temeli algoritmaya dayanır. Aynı zamanda algoritma tek bir problemi çözecek davranışın, temel işleri yapan komutların veya deyimlerin adım adım ortaya konulmasıdır ve bu adımların sıralamasına dikkat edilmelidir. Bir problem çözülürken algoritmik ve sezgisel (herustic) olmak üzere iki yaklaşım vardır. Algoritmik yaklaşımda da çözüm için olası yöntemlerden en uygun olan seçilir ve yapılması gerekenler adım adım ortaya konulur. Algoritmayı belirtmek için; metinsel olarak düz ifade ve akış diyagramı olmak üzere 2 yöntem kullanılır. Algoritmalar bir programlama dili vasıtasıyla bilgisayarlar tarafından işletilebilirler.

Çoğu algoritmalar bilgisayar olarak uygulanmak üzere tasarlanmıştır. Bununla birlikte, başka yöntemlerle de uygulanmaktadır, biyolojik sinir ağı (örneğin insan beyninin hesap yapması veya bir böceğin yemek araması), elektrik devresi veya mekanik cihazlar gibi.

Bilgisayar algoritmasına örnek verelim. Kullanıcının girdiği dört sayının ortalamasını görüntüleyen algoritmayı yazalım:

 A0 --> Başla

 A1 --> Sayaç=0 (Sayaç'ın ilk sayısı 0 olarak başlar.)

 A2 --> Sayı=? : T=T+Sayı (Sayıyı giriniz. T'ye sayıyı ekle ve T'yi göster.)

 A3 --> Sayaç=Sayaç+1 (Sayaç'a 1 ekle ve sayacı göster.)

 A4 --> Sayaç<4 ise A2'ye git. (Eğer sayaç 4'ten küçükse Adım 2'ye git.)

 A5 --> O=T/4 (Ortalama için T değerini 4'e böl)

 A6 --> O'yu göster. (Ortalamayı göster.)

 A7 --> Dur

Quicksort sıralama algoritması animasyonu. Animasyonun başında kırmızı çubukların pivot elemanı olarak işaretlendiği görülmektedir. Örneğin başında sağ tarafın en uzak elemanı pivot olarak seçilmiştir. Bu seçilen pivot değerleri parçalara ayrılmış değerler kümesinin elemanlarıyla karşılaştırılıp sıralama sağlanmaktadır. Özetle, sıralanacak bir sayı dizisini daha küçük parçalara ayırıp oluşan bu küçük parçaların kendi içinde sıralanması ve birleştirilmesi mantığıyla çalışır.

İkinci dereceden ax² + bx + c = 0 biçiminde bir denklemin tüm köklerini bulmak için algoritma yazalım:

Adım 1: Başla.

Adım 2: a, b, c, D, x1, x2, rp ve ip değişkenlerini tanımla.

Adım 3: Diskriminant değerini hesapla.

D ← b2-4ac

Adım 4: Eğer D≥0

x1 ← (-b+√D) / 2a

x2 ← (-b-√D) / 2a

değerlerini hesapla ve x1,x2 değişkenleri göster.

Eğer D≥0 değilse,

Gerçek kısım(rp) ve sanal kısmını(ip) hesapla.

rp ← b / 2a

ip ← √ (D) / 2a

Adım 5: "rp + j(ip)" ve "rp - j(ip)" değerlerini göster.

Adım 6: Dur.

Kullanıcı tarafından girilen bir sayının faktöriyel değerini bulmak için bir algoritma yazalım:

Adım 1: Başla.

Adım 2: factorial,i ve n değişkenlerini tanımla.

Adım 3: Değişkenlerin başlangıç değerlerini tanımla.

factorial ← 1

i ← 1

Adım 4: Ekrandan girilen n değerini oku.

Adım 5: (i=n) eşitliği sağlanana kadar tekrarla.

5.1: factorial←factorial\*i

5.2: i←i+1

Adım 6: factorial değişkeninin değerini göster.

Adım 7: Dur.

3-PSUDO Code

Sözde kod, bilgisayar bilimleri alanında algoritmalar ve programlar oluşturulurken ve aktarılırken kullanılan, günlük konuşma diline benzer ve belli bir programlama dilinin detaylarından uzak anlatımlardır. Programın yapısının ve çalışma mantığının yüksek seviyeli bir biçimde, gerektiği yerde doğrudan doğal dil cümleleriyle, ama yine de bir program yapısı ve akışı içinde anlatılmasıdır. Böylelikle sözde kodu okuyan ya da yazan birisi, programlama dillerinin sözdizim detaylarına dikkat etmek zorunda kalmadan, programın ve algoritmanın çalışma mantığını düşünebilir.

Sözde kod için önceden üzerinde karar kılınmış kesin bir sözdizim yoktur. Sözde kod ile bir programı anlatan kişi, uygun gördüğü programlama dili yapılarının ve işlevlerinin sözde kod içinde bulunduğunu varsayabilir. Amaç, derleme işleminden hatasız çıkacak bir program oluşturmak değil, programın çalışma mantığını anlamak olduğu için, sözde kod yazarken uygun görülen herhangi bir soyutlama düzeyi kullanılabilir. Bazı sözde kodlar programlama dilleriyle büyük ölçüde örtüşürken, bazıları sadece program biçiminde yazılmış düz yazı olabilir.



4- Akış Diagramı

**Akış diyagramı nedir?**

Algoritma bir problemin çözümünün basit, net, sıralı biçimde belirtilmiş halidir. Çözüm için bir algoritma geliştirildikten bu algoritma metinsel, sözde kodlar ile ya da akış diyagramları ile sunulur. Akış diyagramları diğer bir adı akış şeması, algoritmanın belirli şekiller ve simgeler ile ifade edilmiş şeklidir ve algoritma sunumu için yaygınlıkla kullanılır. Bir akış şeması, bir süreç, sistem veya bilgisayar algoritmasını gösteren bir şemadır. Açık, anlaşılması kolay şemalarda karmaşık süreçleri belgelemek, araştırmak, planlamak, geliştirmek ve iletişim kurmak için çok sayıda alanda yaygın olarak kullanılmaktadır.

Akış diyagramları geliştirilen bir algoritmayı şekilsel olarak ifade etmekte ve anlaşılabilirliğini kolaylaştırmaktadır. Akış diyagramlarını gösterir çeşitli şekiller vardır. Bu şekiller işlevlerin simgesel gösterimidir. Bilgisayar programlama dilinde kullanılan programlama kodları bu şekiller ile basit olarak gösterilerek programı yazan programcı yanı sıra başka bir programcı tarafından incelendiğinde anlaşılabilirliğini kolaylaştırmak veya programcı uzun süre önce yazdığı program kodunu unutmuş ise hatırlamasına yardımcı olmak amacına da sahiptir. Otomatik süreci kodlamaya başlamadan önce bir programın arkasındaki mantığı hecelemek için bir akış şemasını kullanabilirsiniz. Büyük resmi düşünmeyi organize etmeye ve kodlama zamanı geldiğinde bir rehber sağlamaya yardımcı olabilir.

Genel olarak bilgisayar programlama dillerinde programlamaya başlarken kodlar yazılmadan önce belirlenen algoritma akış diyagramları ile ifade edilir. Akış diyagramları belirlenen bilgisayar programlarının kaynak kodlaması yapılır. Bu işlem programın kaynak kodlarının geliştirilmesi için gerekli olan öngörü ve işlem kolaylığını sağlar. Akış diyagramı oluşturulan bilgisayar programında algoritmanın nasıl uygulanacağının bilinmesi veya daha kısa bir yolu var ise bu şekilsel olarak görülmesi konusunda yardımcı olmaktadır. Algoritmalar bir iş veya işlemin nasıl yapılacağını gösteren yazı veya şekillerden oluşur.

Aşağıda gösterilen akış şeması şekilleri ve sembolleri, çoğu akış şeması diyagramında bulabileceğiniz en yaygın türlerden bazılarıdır.

|  |  |
| --- | --- |
| Bağlantı Sembolü  Akış yönü ve semboller arasındaki bağlantıyı göstermek için kullanılır. | https://lh5.googleusercontent.com/tKyXATpXJqDT-1WN_DLGRcgQ8I97d7UgI7dA3m2JE481KhagaGzdjB-2SNuGYE9df4w0ZnREfhl6cc548K3bjIP0H2iE84VCMytyXI8k7o-JCfR61MybfG7OYIPdQhki0I6P3oSMpQytz8TEBQ |
| Başlangıç ​​/ Bitiş Sembolü  Sonlandırıcı, işleminizin nerede başladığını veya bittiğini gösterir. İşleri daha açık hale getirmek için sonlandırıcı şeklinin içinde ‘Başlat’, ‘Başla’, ‘Son’ gibi kelimeleri kullanabilirsiniz. | https://lh3.googleusercontent.com/AUtq-1eAGVGfqnLJfxy_0y8gXfg_otZxAqUoDNRQ6sDcR-AXfQFaieYhct8zaYC_HmhOM-o3kN_0Lf2ZevcYqsOe-ljbm7nJdd-OJbq-LaScyW7V7Z6k9YbV9aZNvAhf4rU4K51cBtGhIYLpXg |
| Eylem veya Süreç Sembolü  Bir işlemi, süreci veya aktiviteyi ifade eder | https://lh4.googleusercontent.com/mDZmyyUI43WXmIBnE3BDsJ973lfZNqvLs2rv6EG1pOWOu7U1ltbPovw_6tGIAC9EW_EOh__SLO-YiLJaXxuV6gyaHhGW7v1JcinXIBRrqfvlVOafzRg04V9Ea3oTnXIUgm6GAJA901JEqeNyeg |
| Belge Sembolü  Basılı bir belge veya raporu temsil eder. | https://lh4.googleusercontent.com/IUbxmRWDx6H4JNFdkisuyjIcC7xMc5JI3UxSNYsCi4qrrbBQCqRz5RZ4pZFsU1aajQJul1LNOpMk9PglZZ3-9IH_ycbeRkW4FPgR4tGfqOjF_eGXxf9HYJevFt44JDOW22CRuHtzPyYwf-dMoQ |
| Birden Fazla Belge Sembolü  İşlemde birden çok belgeyi temsil eder. | https://lh5.googleusercontent.com/0jC3GF6keOvWZy2Rcey85nIhYXMNXlMBeMEOaQ4z51sGVoP_k7ulw9J2WhzJnryOjQp7FDiph9O9oKMew_TfpOCO5PjJ0U5KFL_9YRkqaXSxy-2JBXs6Z1U0ppjOealSIRp1_n9wHUxyxl9eSA |
| Karar Sembolü  Bir karar veya dallanma noktasını temsil eder. Bir kararın alınması gerektiğini ve bu karardan dolayı süreç akışının belirli bir yönü izleyeceğini göstermektedir. | https://lh5.googleusercontent.com/_9YIT1FEr7NIrRvy82lE6qJ3UQu5k1tgi1FnWXpeGe92aC_1ddonjJP5fGIEZ020FU0k1WTO1ccXp8mpQPk-pSteInkBpRynrdc2768HsXY4f5xo3_tqiZq3frWP4pyBRui-LSrlp81FFWk7DQ |
| Giriş / Çıkış Simgesi  Müşteri siparişi (giriş) veya ürün (çıktı) gibi sisteme giren veya sistemden gönderilen materyal veya bilgileri temsil eder. | https://lh4.googleusercontent.com/Q0byh_Caj4ObY4L8aUq4nR48H-svxX9tm0fadoZDaKVkpkDDA2qHQyHfAFQOcRmKBOOZCprgMLEhVmYPw24KIyZguX8bYHgrmXf-ErvvHONxOtN6SG8LzBUXBiEp6c8Ifz0BvPAkWlEB_YG0Og |
| Manuel Giriş Sembolü  Kullanıcının bilgileri manuel olarak girmesi istendiği bir adımı temsil eder. | https://lh5.googleusercontent.com/61MAG6dyNYcsCUxizDQg8TgnZYkC-vEMCEal1nbgJkGS_nk-CcKtEB8725OaP4ehCndnczpvMAg4XaXiqA5N_elmnG7qPq9BnSZhGEaGbBTFIHWI23_JsKENjyMPnhFBx146pFH4FXynXRkdIw |
| Hazırlık Sembolü  Süreçte başka bir adımın ön hazırlık veya kurulumunu temsil eder. Devam etmeden önce bir işlemin yapılması, ayarlanması veya değiştirilmesi gerektiğini gösterir. Kurulumun aynı süreç içerisinde başka bir adıma geçmesine yardımcı olur. | https://lh6.googleusercontent.com/j2O-8m8iWPUZBxuw5NnxZ5vo0LITT788baIv81ejHqdl84CoefI73RaGe08blM-qE3BFGTSBey_n1JmSizppV82NVxvm6w9GFh2Ja4ehOnw8Rluz5RJgDGRDCG2T46nk76eoCVKg-oWFPWAsHA |
| Bağlayıcı Sembolü  Akışın, eşleşen bir sembolün (aynı harfi içeren) yerleştirildiği yerde devam ettiğini gösterir. | https://lh5.googleusercontent.com/AV1ZuHC8ZDga7-RgQh-tOKYB3ishJ7LwNGBm7eg9l7p2jmv-Jpf4GoyrlAZRulIhyHe_I8TBeThrNoeMidNu1VEedv7uYWTKpIv0-AOtfIVYGju1FdRtRHWRPGVMzZdbIGu64teRVrjRqVaVnQ |
| Veya Sembolü  Proses akışının ikiden fazla dalda devam ettiğini gösterir. | https://lh5.googleusercontent.com/gbkVNGvVy9Cf7-iWHadHsr0ktfXROP_j1tc9glRL4fzXJrRskNnucXFdRM6o6ugOUKiLR8_H3beBnh-aD3_EYEVHKPPzvlaMCKW2uN1D1HhSRgdJ1hM737JWhKOHscJZInKU51kwUlPKIF4lVw |
| Alt Yordam sembolü  Daha büyük bir süreç içinde gömülü belirli bir görevi gerçekleştiren eylemler dizisini belirtir. Bu eylem dizisi, ayrı bir akış şeması üzerinde daha ayrıntılı olarak tarif edilebilir. | https://lh4.googleusercontent.com/1I3xOzxliCx-nLqcH6ApvCDzgnUpIU4a9tAD2bueeIrDStfH-L8PdUDAbxjdXefpLZ9fJ5wHdAECjuC7f7FkiSTi26JJL6wQJ__PidctNQm6yuTlprXWTjVCZvl6tFo2vOnku_sOui3PwFjrwg |
| Manuel Döngü Sembolü  Manuel olarak durdurulana kadar tekrarlamaya devam edecek bir dizi komut dizisini belirtir. | https://lh4.googleusercontent.com/SQy8Pkqgj5rKisNOVEdjVvE0U14OFAMyO3Dgr6oe6CJGsUUqdSEqZUCsI_EhsLuV_5cdqJHNtILY_yrwRnv4kUbX_Rw43-oV_Kzq0gVtbteLHdd2m1qxDiCeqeo0P1eKDmIHFC2N_0JcnDxf9Q |
| Döngü Sınırı Sembolü  Bir döngüün durması gereken noktayı belirtir. | https://lh5.googleusercontent.com/KEWZ18mC-auUOtbjjdMjxC7_NgwJ8vFprZDM-PGxEgPs9Wp3JY0fchq-jgSfAUTgp28yPHusUotXNrg5HKibCVqkngxTW-E_O00laTMnMH33TXNjxtdFGSOtxwuXRwT-D66DOz9BAdOHUm6Kew |
| Gecikme Sembolü  İşlemde gecikme olduğunu gösterir. | https://lh4.googleusercontent.com/-LP1Wer5UDANdZUWtGTRauCmkUgWHpRJrmEdZea1jSw8wZ8a0slz1hcw6vMpunAjgT3FIL6eVvxrBzo1gUkyQLmRyQ3LdjN2U1X7RVju6UdWu7HNHlSeQzTwk8y5Nqa6Zy7xRj02izEhiskkRA |
| Veri Depolama veya Saklanan Veri Sembolü  Verilerin saklandığı bir adımı belirtir. | https://lh6.googleusercontent.com/Iu9EcQtl283jn4Ppf4qAZkSQKCyq1-OGUew_w0mtYBDEQr71sL-Can-FB8G-c39zovDahPY5cL36H0vukRsg6KAfZFUIBGdz3fWPM_uJaCVCPDwNBS3N2Eq8hmPZ-gtJhsrvjsVhAXFZWRL0jg |
| Veritabanı sembolü  Arama ve sıralama için izin veren standart bir yapıya sahip bilgi listesini gösterir. | https://lh3.googleusercontent.com/WZDK7MawbSADp6ayxtlStCf9uPc4ousfB9ZJKMjKryEUHzHMva7KLdastAovERbZHtfKNgz-ALrCo_CheFbIQdGqMlpfbnNc03kXboLrm9aa0DLpl3BlrrIWJEqEnWJrif429mKnfPduN0jojw |
| Dahili Depolama Sembolü  Yazılım tasarım akış şemalarında kullanılan bir program sırasında bilgilerin bellekte saklandığını belirtir. | https://lh4.googleusercontent.com/k23wPG5PrDyHUunjr_NTPbAeqPaJZpf9d0wjdwq7nw0SPLU524H5_4qBH2JMcnh7tku7F1TgZnhWhTAJAcGU7Yvp-DLyjbcDXcgrJQ2ervah7E2_6nLiJnb-PIDEXoDyCHbs34B3gisReb6geg |
| Ekran Sembolü  Bilgileri gösteren bir adımı belirtir. | https://lh6.googleusercontent.com/wyjCaZ5tIE5WA2WAMiZ9ouKI25DWoXAMFdcTNNcVER8xf4CwM_DvGP5M3_a6TOzPB25KDfiMTLylSczE4FhbLFcfT-RQl_BtbwSH_un0HKvdQgEDEa1LdKVvzUdeu-6_p20P25v71uEYwhP85A |

Temel bir akış şeması nasıl planlanır ve çizilir

1. Amacınızı ve kapsamınızı tanımlayın. Başarmayı umduğun şey nedir? Bu amaca ulaşmak için uygun başlangıç ​​ve bitiş noktaları ile doğru şeyleri mi çalışıyorsunuz? Araştırmanızda yeterince detaylı olun, ancak planladığınız kitlenizle iletişim kurmaya yetecek kadar basit tutmaya özen gösterin.
2. Görevleri kronolojik sırada tanımlayın. Bu, uygulamanızda rol alan katılımcılarla konuşmayı, bir süreci gözlemlemeyi ve / veya mevcut belgeleri gözden geçirmeyi içerebilir. Not formundaki adımları yazabilir veya kaba bir tablo ile başlayabilirsiniz.
3. Bunları süreç, karar, veri, girişler veya çıktılar gibi tür ve karşılık gelen şekle göre düzenleyin.
4. Grafiğinizi çizin, ya elle çizim yapın ya da özel bir program kullanın.
5. Akış çizelgenizi doğrulayın , sürece katılan kişilerle adım adım ilerleyin. Amacınız için önemli olan hiçbir şeyi kaçırmadığınızdan emin olmak için süreci gözlemleyin. Üzerinde herkesin katıldığı genel testler yapın.