

BAŞİSKELE GÜBRETAŞ ORTAOKULU 2024-2025



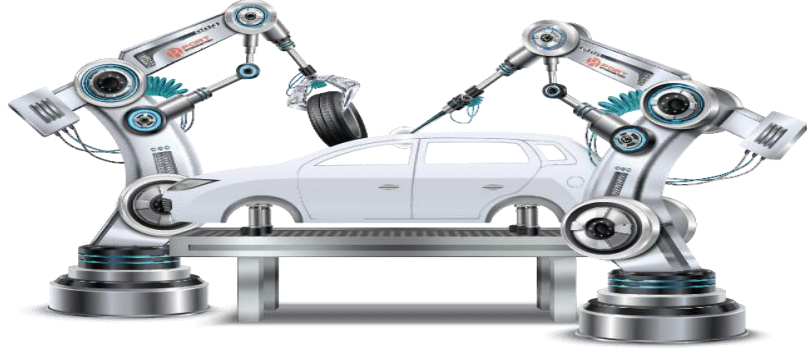
EĞİTİM-ÖĞRETİM YILI

ROBOTİK KODLAMA

İÇİNDEKİLER:

- 1) Robotik nedir?
- 2) Robotik uygulamaların günlük hayata katkıları, olumsuz etkileri ve riskli yönleri.
- 3) Yazılım, problem, algoritma, akış şeması.
- 4) Programlamanın Temelleri
 - a) Döngüler
 - b) Karakteri Hareket Ettirme, başlatıcılar
 - c) Karakterin Yönü
 - d) Sahnede konum
 - e) Karakteri yön tuşları ile hareket ettirme
 - f) Karar-Mantık yapısı
 - g) Operatörler
 - h) Değişkenler
 - i) Sayaçlar
 - j) Projeler

1. Robotik nedir?



Robotik, robotların tasarımı, geliştirilmesi, programlanması ve kullanımıyla ilgilenen disiplinler arası bir bilim ve mühendislik alanıdır. Robotik, genellikle bilgisayar bilimi, elektrik mühendisliği, makine mühendisliği ve yapay zeka gibi alanların birleşiminden oluşur. Bu disiplin, fiziksel ve dijital dünyalar arasında bir köprü oluşturur ve robotların çevreleriyle etkileşim kurabilmesi, karar verebilmesi ve belirli görevleri otonom olarak yerine getirebilmesi için gerekli olan donanım ve yazılımı kapsar.

Robotik, şu alt alanları içerebilir:

1. Mekanik Tasarım: Robotların fiziksel yapısının tasarlanması (eklemler, kollar, tekerlekler vb.).
2. Kontrol Sistemleri: Robotların hareketlerini ve görevlerini yerine getirmek için kullanılan elektronik sistemler.
3. Algılayıcılar (Sensörler): Robotların çevresini algılayabilmesi için kullandığı sensörler (kamera, radar, lidar gibi).
4. Yapay Zeka ve Makine Öğrenimi: Robotların çevrelerine uyum sağlamasını ve otonom kararlar almasını sağlayan yazılım ve algoritmalar.
5. Hareket Planlaması: Robotların belirli görevleri en uygun şekilde yerine getirmek için yollarını planlama yeteneği.
6. Etik ve Sosyal Yönler: Robotların insanlarla olan etkileşimleri, etik kullanımı ve sosyal etkileri.

Robotik, günümüzde endüstriyel üretim, sağlık hizmetleri, tarım, lojistik, uzay keşfi gibi birçok alanda yaygın bir şekilde kullanılmaktadır.

2) Robotik uygulamaların günlük hayata katkıları, olumsuz etkileri ve riskli yönleri.

Günlük Hayata Katkıları

Robotik uygulamalar, günlük yaşamda birçok alanda fayda sağlar. İşte öne çıkan katkıları:

Sağlık Hizmetleri: Robotlar, cerrahi işlemlerde yüksek hassasiyet sunarak ameliyatların daha güvenli olmasını sağlar. Hastalara ilaç ulaştırmak, taşımak veya fizyoterapi desteği sağlamak için de kullanılırlar.

Ev Otomasyonu: Ev işleri için robot süpürgeler veya pencere temizleme robotları gibi cihazlar yaygınlaştı, insanlara zamandan tasarruf sağlıyor.

Engelliler için Yardımcı Teknoloji: Robotlar, hareket kabiliyetini sınırlayan bireylere bağımsız yaşam desteği sunar, günlük işlerini yapmalarına yardımcı olur.

Tarım: Otonom traktörler, ekim, ilaçlama ve hasat gibi işlemlerde verimliliği artırarak çiftçilere büyük kolaylık sağlıyor.

Güvenlik: Güvenlik robotları, tehlikeli bölgeleri gözetim altında tutarak riskleri azaltır, özellikle sanayi tesislerinde ve halka açık alanlarda güvenlik sağlar.

Eğitim: Eğitim robotları, çocuklara kodlama, matematik ve bilim gibi konuları eğlenceli bir şekilde öğretir, öğrenme sürecini interaktif hale getirir.

Perakende ve Müşteri Hizmetleri: Otonom raf düzenleyiciler veya müşteri karşılayan robotlar, mağaza operasyonlarını daha verimli kılar ve müşteri deneyimini iyileştirir.

Lojistik ve Depolama: Depolarda yük taşıma ve yerleştirme işlerini hızlandırarak insan çalışanların yükünü azaltır ve işletmelerin daha verimli çalışmasını sağlar.

Yolculuk ve Ulaşım: Otonom araçlar, yolculukları daha güvenli ve çevre dostu hale getirme potansiyeline sahiptir. Taksi ve teslimat hizmetlerinde de kullanılmaktadır.

Afet Yardımı: Doğal afetlerde, kurtarma operasyonları için robotlar kullanılarak tehlikeli alanlarda insanları kurtarmaya yardımcı olunur.

Eğlence: Animatronikler ve robotik oyuncaklar, eğlence sektöründe yeni deneyimler sunar.

Bu alanlarda robotlar, iş yükünü azaltırken kaliteyi ve güvenliğini artırmakta, yaşam kalitesine katkıda bulunmaktadır.

Olumsuz Etkileri ve Riskli Yönleri

Robotik uygulamaların günlük hayata getirdiği pek çok fayda bulunmakla birlikte, bazı olumsuz etkileri de vardır. İşte bunların başlıcaları:

İş Kaybı: Otomasyon nedeniyle birçok sektörde insanlar işlerini kaybedebilir. Özellikle üretim ve lojistik gibi alanlarda robotların kullanımı, insan iş gücünü azaltmaktadır.

Ekonomik Eşitsizlik: Robot teknolojisine yatırım yapabilen büyük şirketler, rekabet avantajı kazanırken küçük işletmeler geri planda kalabilir. Bu durum ekonomik eşitsizlikleri artırabilir.

Bağımlılık ve Teknolojiye Güven: Robotlara aşırı güven, insanların becerilerinin azalmasına ve kritik kararları robotlara bırakma eğilimine yol açabilir.

Mahremiyet ve Güvenlik Sorunları: Güvenlik ve gözetim robotları, bireylerin mahremiyetine müdahale edebilir. Kameralı robotlar veya yapay zeka destekli izleme sistemleri, veri güvenliği riskleri oluşturur.

İnsan-Robot İletişimi Zorlukları: Robotların insanlarla etkili iletişim kurması halen zor bir konudur. Yanlış anlamalar, iş yerlerinde veya evlerde sorun yaratabilir.

Yüksek Maliyet: Robot teknolojileri, özellikle başlangıç aşamasında oldukça pahalıdır ve her bütçe için erişilebilir olmayabilir. Bu, toplumsal eşitsizliği artırabilir.

Etik Sorunlar: Robotların insan kararlarını etkilemesi, etik soruları beraberinde getirir. Örneğin, otonom araçların kaza durumunda nasıl davranacağı gibi konular etik tartışmalara yol açmaktadır.

Yapay Zeka Yanlılıkları: Robotların karar verme süreçlerinde kullanılan yapay zeka algoritmaları, önyargılı sonuçlar doğurabilir, bu da ayrımcılığa yol açabilir.

Kültürel ve Sosyal Değişimler: Robotların yaygın kullanımı, toplumun geleneksel yapısını ve sosyal normları değiştirebilir, bazı kültürel değerlerin kaybolmasına yol açabilir.

Robotik uygulamaların günlük yaşamda kontrollü bir şekilde kullanılması, bu olumsuz etkileri minimize etmek adına önemlidir.

Sosyal İzolasyon: Sosyal robotlar ve yapay zeka asistanları, insanlar arası etkileşim ihtiyacını azaltarak sosyal izolasyonu artırabilir.

Hataların Sonuçları: Robotlar her ne kadar programlanmış olsa da, teknik hatalar yapabilirler. Özellikle sağlık ve güvenlik alanında kullanılan robotların hata yapması ciddi sonuçlar doğurabilir.

Fiziksel Güvenlik Tehditleri: Ağır işlerde kullanılan robotların yanlış çalışması veya arızalanması, çevresindeki insanlar için tehlike oluşturabilir.

Psikolojik Etkiler: İnsanların robotlarla etkileşime girmesi, uzun vadede psikolojik etkiler yaratabilir; robotlara aşırı bağlanma veya soğuk ve mekanik ilişkilerden hoşnutsuzluk gibi.

Çevresel Etki: Robot üretimi, bakımı ve atık yönetimi, çevre üzerinde olumsuz etkiler yaratabilir. Üretim sürecinde kullanılan malzemeler çevresel kirliliğe neden olabilir.

Yeteneklerin Gelişmemesi: Özellikle çocuklar, günlük işlerini robotlara yaptırdıklarında, pratik becerilerini geliştirmede geride kalabilirler.



YAZILIM: Çeşitli görevleri gerçekleştirmek amacıyla hazırlanmış programlara **yazılım** adı verilir. Her yazılım bir **problemi** çözmek amacıyla geliştirilmiştir.



PROBLEM: çözülmesi gereken sorun ya da aşılması gereken engel anlamına gelir.

Bir Problemin Çözümü İçin:

- Problemi iyi anlamak
- Sonucun doğruluğunu kontrol etmek.
- Kısa ve anlaşılır şekilde çözmek.



Günlük yaşamda karşılaştığımız problemleri bilerek veya farkında olmadan adım adım çözmeye çalışırız.

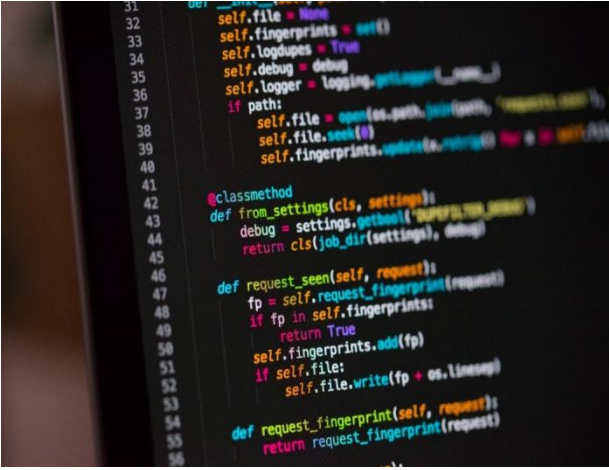
Örneğin yazı yazarken kaleminizin ucu kırıldığında şu adımları takip ederek bu sorunu çözersiniz.

1. Kalem tıraşı çıkar.
2. Kalemi al.
3. Çöp kovasının yanına git.
4. Kalemin ucunu aç.
5. Sırana geri dön.
6. Yazmaya devam et.

Bilgisayarlar da problemleri tıpkı bizler gibi adım adım çözmeye çalışır.

Kullandığımız yazılımların tamamı «**kod**» adı verilen bilgisayarın anlayacağı dilde yazılmış özel komutlardan oluşur.

Bu kodlar bilgisayar yazılımcıları tarafından yazılır.



```
31 self.file = None
32 self.fingerprints = set()
33 self.logdupes = True
34 self.debug = debug
35 self.logger = logging.getLogger(__name__)
36
37 if path:
38     self.file = open(os.path.join(path, "requests.log"),
39                     "a")
40     self.fingerprints.add(fp)
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool("DEBUG")
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```


Kurt, Kuzu, Ot Problemi



- Tekneye her defasında sadece 1 tanesi alınabiliyor.
- Yani ya kurdu, ya kuzuyu yada otu almanız gerekiyor.
- Eğer kurt ile kuzuyu baş başa bırakırsanız, kurt kuzuyu yer.
- Eğer kuzu ile otu baş başa bırakırsanız, kuzu otu yer.

Çözümü yapalım: <https://www.rekoroyun.com/kurt-kuzu-ot.html>

Kurt, Kuzu, Ot Problemi Çözümü

1)

2)

3)

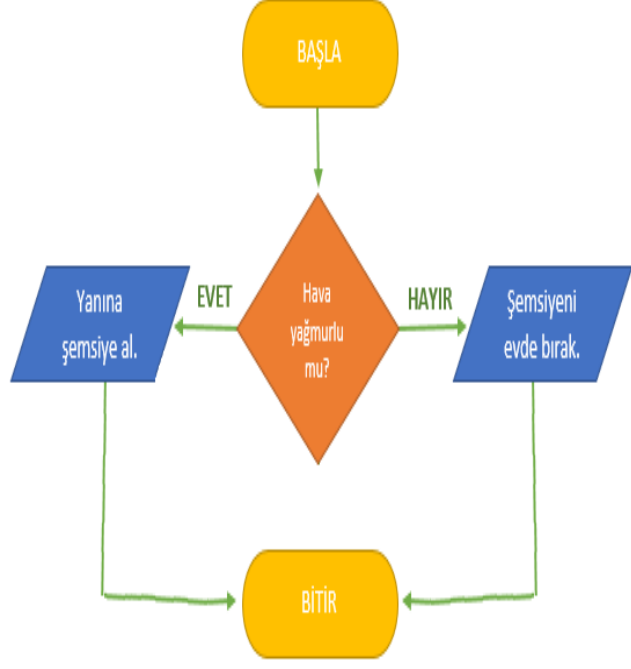
4)

5)

Kodlamaya başlamadan önce oluşturacağımız yazılımın adım adım ne yapacağını tasarlamamız gerekir.

İşte açık ve net ifadelerle problemin adım adım çözümünü gösteren bu taslağa «**algoritma**» adı verilir.

Programlamanın ilk adımı algoritma oluşturmaktır.



ALGORİTMA: Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur.

Algoritma basamaklarının bir başlangıcı ve sonu bulunur.

Her adımda yapılacak işlem açıkça belirtilir.

Oluşturacağımız yazılımın kusursuz olması için öncelikle her adımını gösteren planını, yani algoritmasını hazırlamalıyız.

AKIŞ DİYAGRAM ŞEKİLLERİ



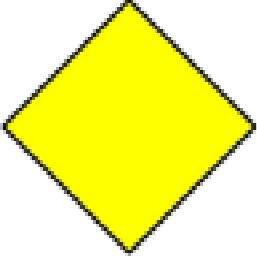
Programın başlangıç ve bitişi için kullanılır.



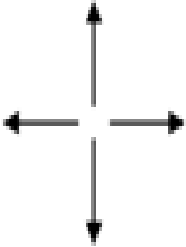
Bilgi giriş çıkışı için kullanılır.



Aktarma, aritmetik hesaplama, işlem



Karar alma için kullanılır.



Birleştirme çizgileri

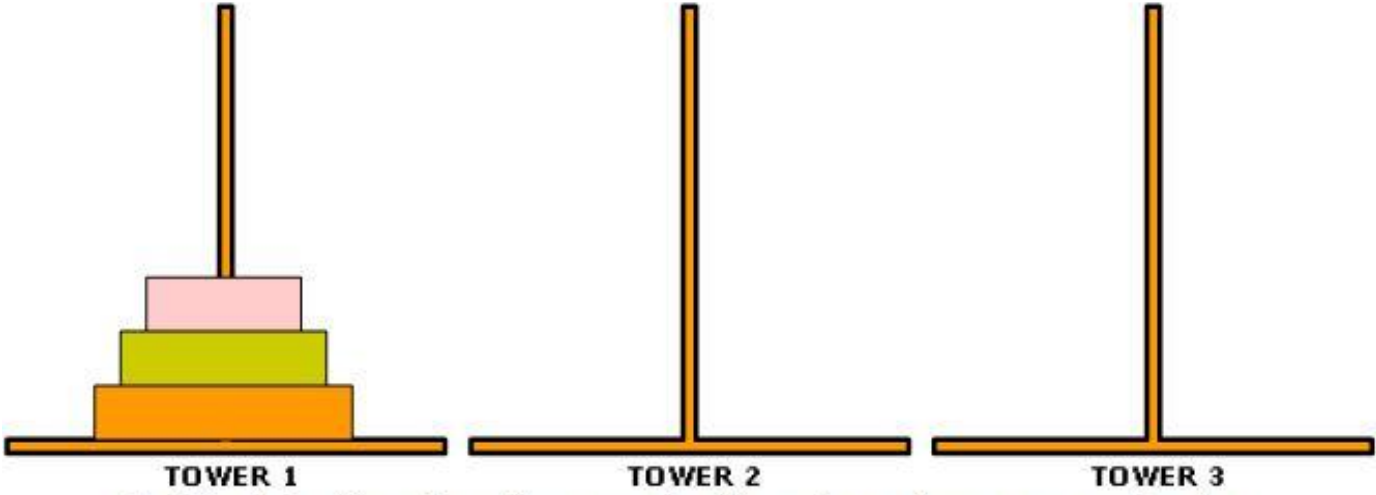


Yazdır



Bağlantı

Hanoi Kuleleri



O objetivo do jogo é transferir os discos para a torre três, porém um disco menor nunca deve ficar em baixo de um maior. Para movimentar um disco segure-o com o mouse e transfira para a torre.

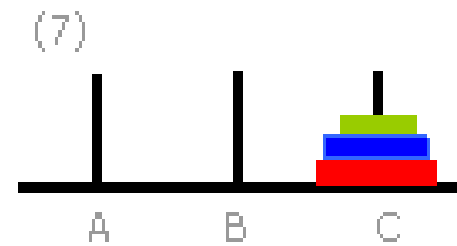
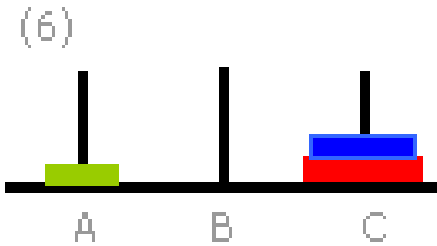
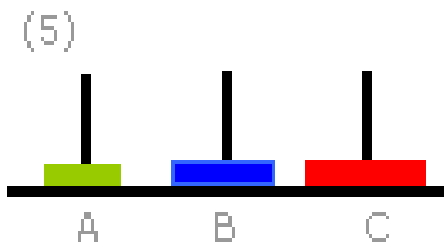
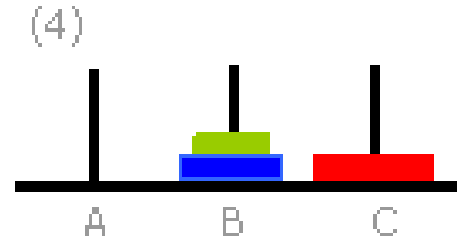
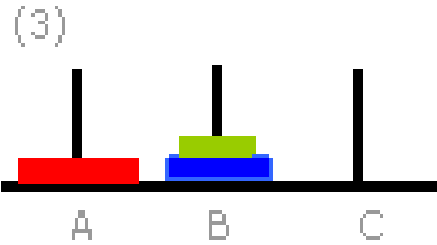
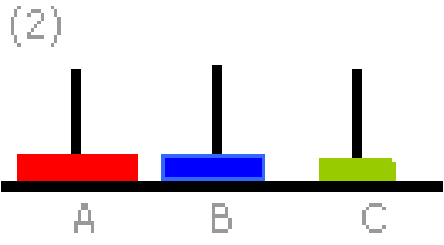
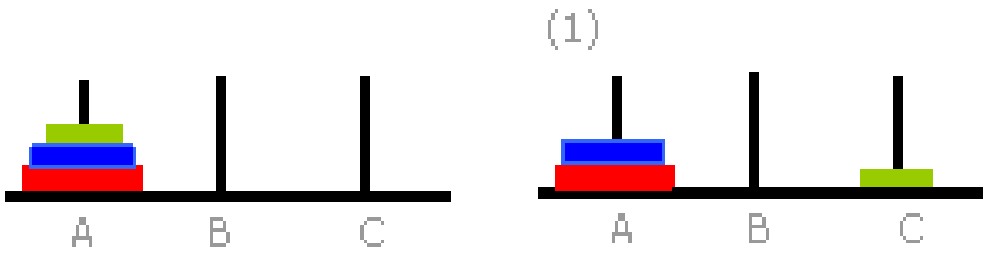
Discs: Moves: Min:

Problemi çözelim: <https://www.rekoroyun.com/hanoi-kuleleri.html>

- Hanoi kuleleri oyununda amaç; En sol taraftaki kulede bulunan diskleri, en sağ tarafta bulunan kuleye büyükten küçüğe doğru olacak şekilde yerleştirmektir.
- Küçük disklerin üstüne kendinden büyük disk koyamazsınız.
- Oyun en az 3 en fazla 6 disk ile oynanmaktadır. Disk sayısı arttıkça oyun zorlaşmaktadır.
- Disk sayısı arttıkça minimum hamle sayınızda buna göre değişmektedir. *Ör: 3 diskte sizden 7 hamlede yapmanız istenirken, 6 diskte ise 63 hamlede tamamlamanız istenmektedir.*
- Oyunda geri alma yoktur. Bu yüzden hamlelerinizi dikkatli yapınız.
- Her hamlede sadece 1 parçayı taşıyabilirsiniz.

Hanoi Kuleleri Çözüm

3 Disks



- 1) Yeşili C'ye götür.
- 2) Maviyi B'ye götür.
- 3) Yeşili B'ye götür.
- 4) Kırmızıyı C'ye götür.
- 5) Yeşili A'ya götür.
- 6) Maviyi C'ye götür.
- 7) Yeşili C'ye götür.

Algoritmayı test edelim:

<https://www.rekoroyun.com/hanoi-kuleleri.html>

Hanoi kuleleri oyununun 4 diskli olan bölümünü çözerek problemin algoritmasını liste halinde yazalım.



- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)
- 8)
- 9)
- 10)
- 11)
- 12)
- 13)
- 14)
- 15)

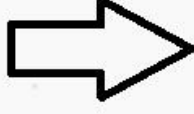
PROGRAMLAMANNIN TEMELLERİ

a) Döngü: Bir veya birden fazla işlemi, bir koşula bağlı olarak, belirli sayıda veya bir koşul sağlandığı sürece tekrarlayarak çalıştıran kalıplara **döngü** adı verilir.

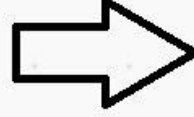
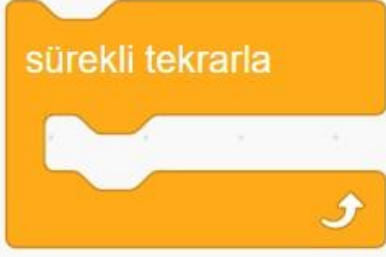


Örnek: 10 adet İLERLE bloğu yerine TEKRARLA bloğu kullanarak 10 kez tekrarla yazılır ve içerisine 1 tane İLERLE bloğu koyularak işlem tamamlanır. Bu sayede 10 blokta yapılacak program 2 blok ile tamamlanır.

DÖNGÜ TÜRLERİ

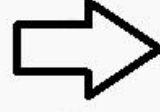


Yazılan sayı kadar içerisinde ki blokları tekrarlatır.



Program durdurulana kadar içerisinde ki blokları tekrarlatır.

Koşul



Yazılan koşul gerçekleşene kadar içerisinde ki blokları tekrarlatır. Koşul gerçekleştiği anda tekrarlatmayı durdurur.

b) Karakteri Hareket Ettirme, Başlatıcılar



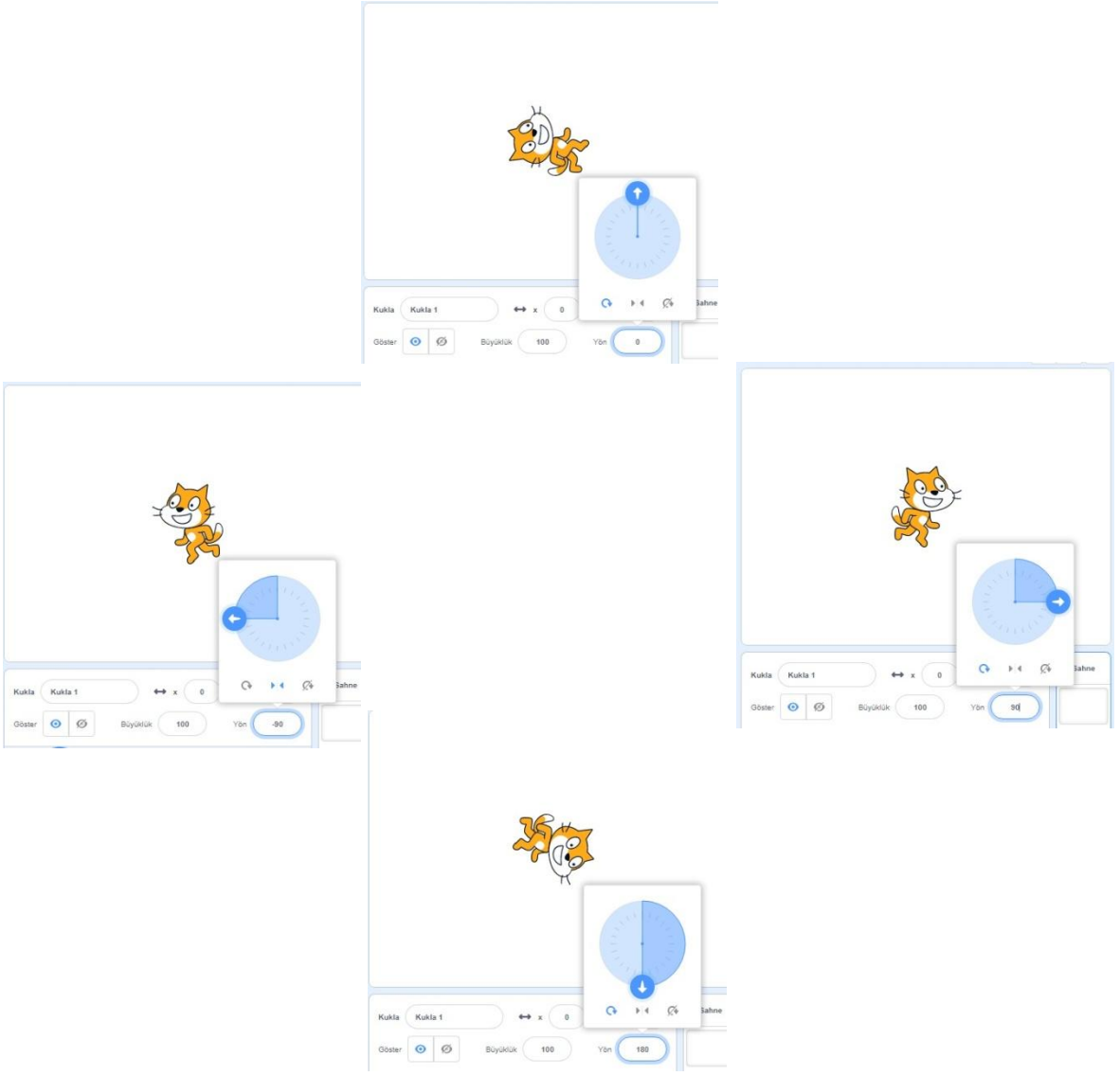
Yeşil bayrak tıklandığında oyun başlar.

Program durdurulana kadar sürekli tekrarlar bloğu çalışır.

Her çalıştığında sonraki kostüme gelir.

Karakter sahnenin köşesine geldiğinde geri döner.

c) Karakterin Yönü



Karakterimizin sahnede hangi yöne bakacağı 0 – 180 derece ve 0 - -179 derece arasında belirlenir.

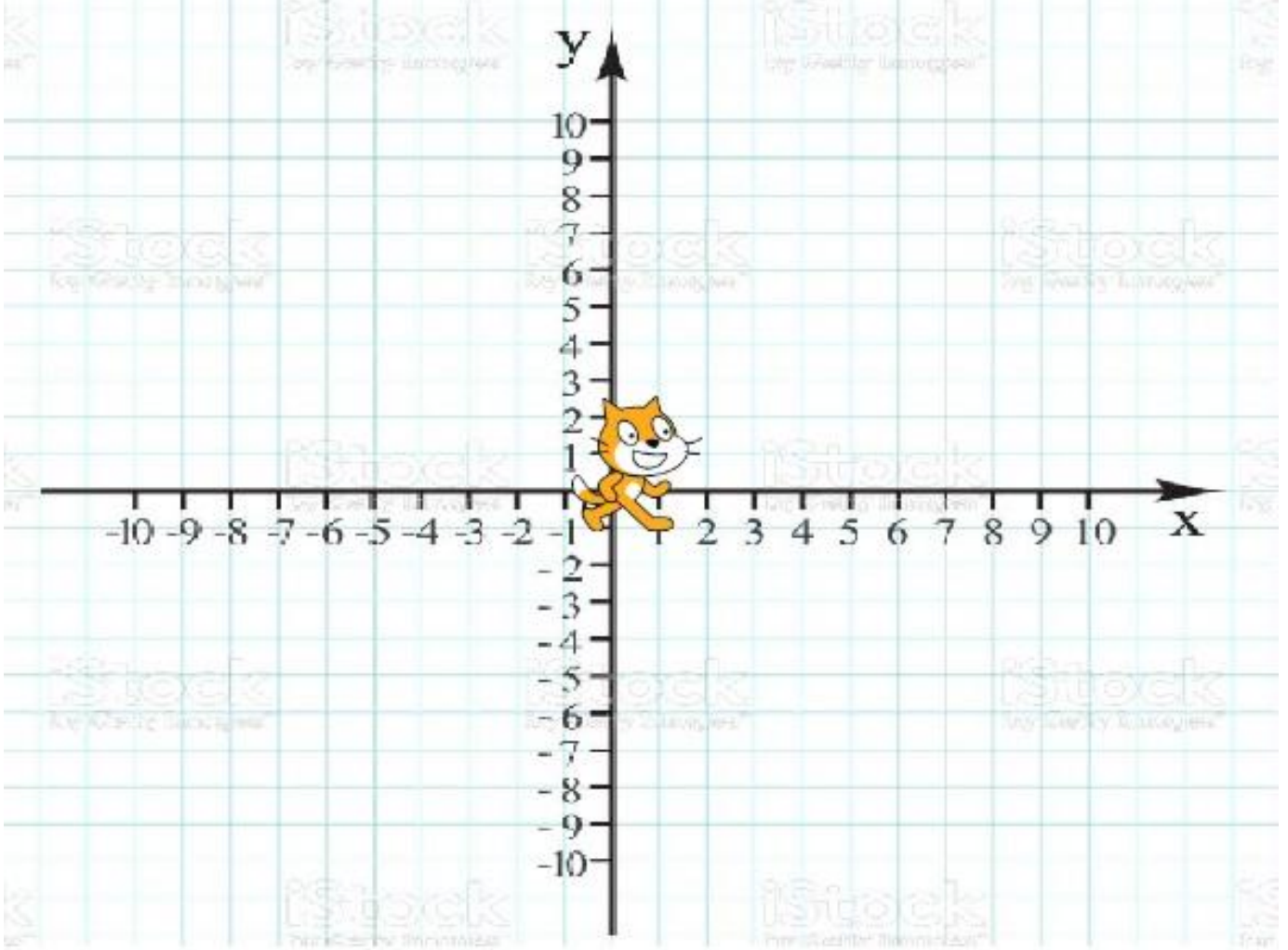
0 derece ekrana göre yukarı bakar.

90 derece ekrana göre sağa bakar.

180 derece ekrana göre aşağı bakar.

-90 derece ekrana göre sola bakar.

d) Sahnede Konum



Sahnede yer alan karakterimizin konumu önce x eksenine, sonra y eksenine göre belirlenir.

Tam orta noktasında olmasını istiyorsak (0,0) noktasına getirmeliyiz.

X değeri artarsa karakterimiz sağa, azalırsa sola hareket eder. Y değeri artarsa karakterimiz yukarı, azalırsa aşağı yönde hareket eder.

Çapraz yönlerde hareket ettirmek istersek x ve y değerlerini aynı anda artırmalı veya azaltmalıyız.

e) Karakteri Yön Tuşları İle Hareket Ettirme

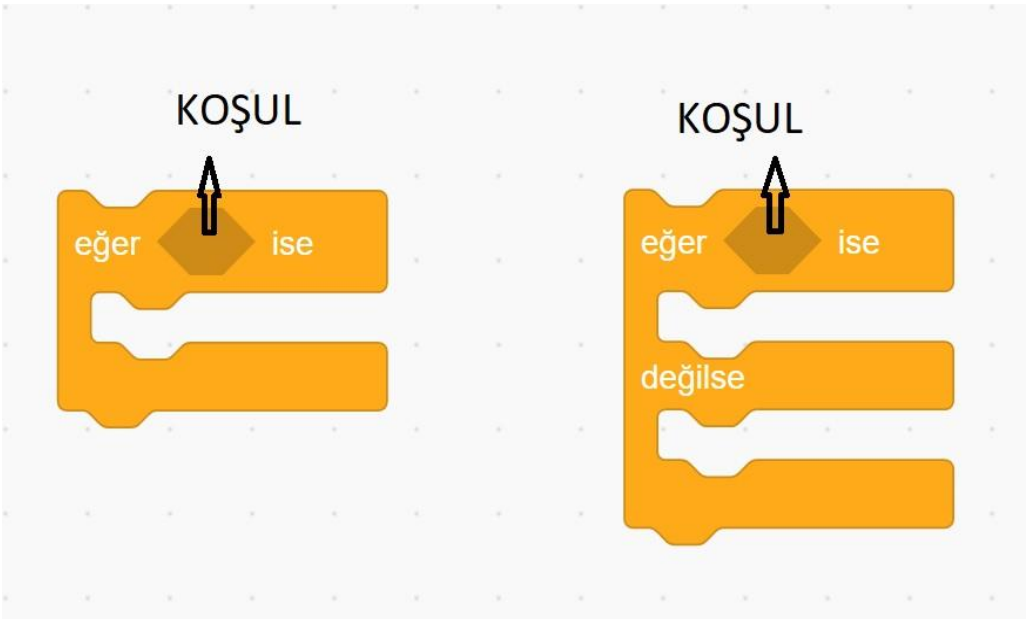
yukarı ok ▼ tuşuna basılınca
y konumunu 10 değiştir

sol ok ▼ tuşuna basılınca
x konumunu -10 değiştir

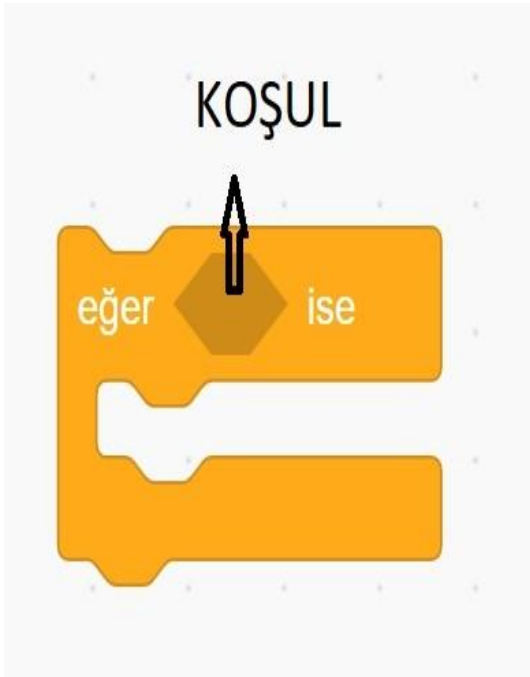
sağ ok ▼ tuşuna basılınca
x konumunu 10 değiştir

aşağı ok ▼ tuşuna basılınca
y konumunu -10 değiştir

f) Karar Mantık Yapısı



- Bilgisayara iki yada daha fazla seçenek arasından seçim yaptırmak için kurulan mantık yapılarıdır.

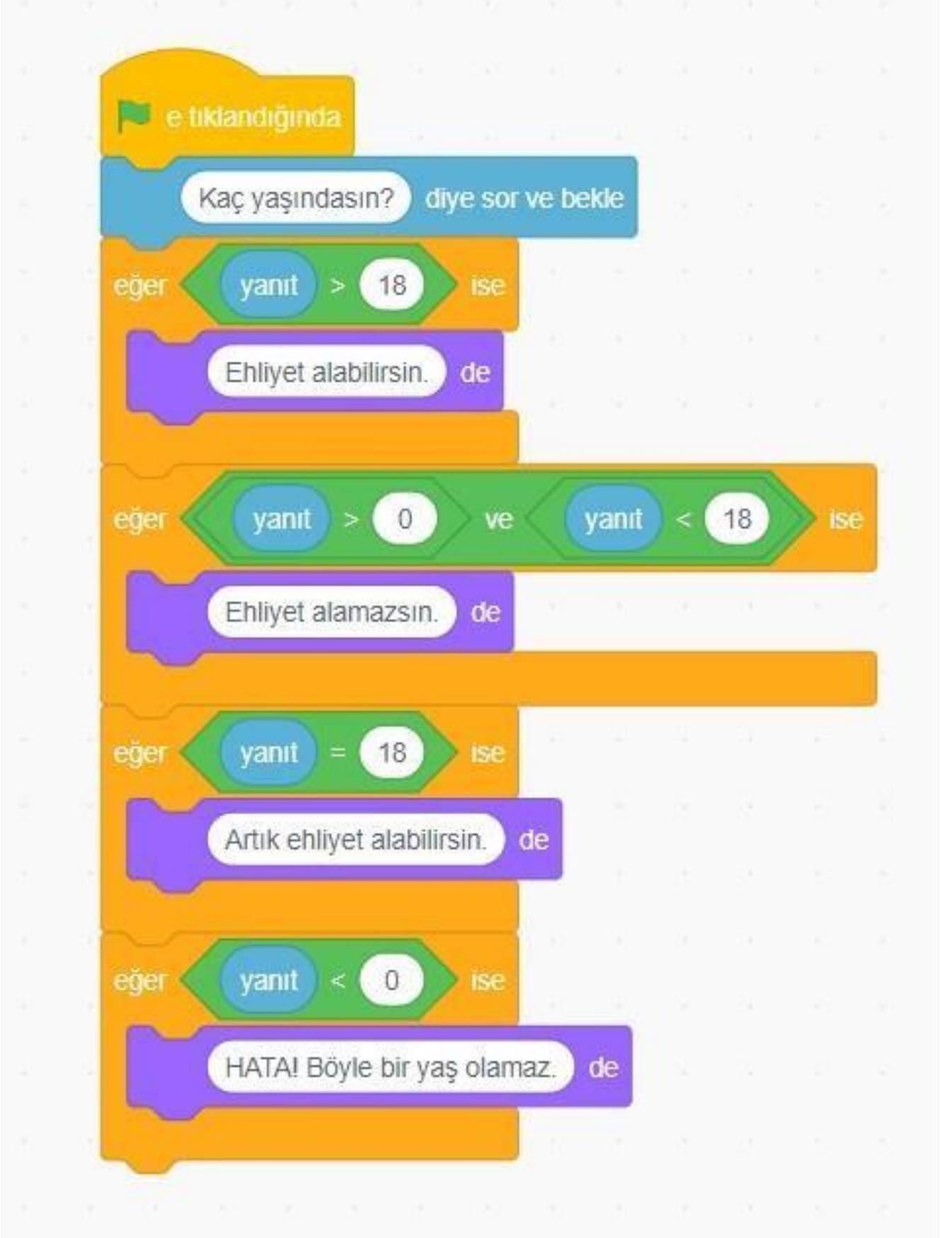


Yazılan koşul **doğru** ise içerisindeki blokları çalıştırır, **yanlış** ise blokları çalıştırmadan alttaki komutlardan devam eder.

Koşul gerçekleşir ise ilk boşlukta yer alan bloklar çalışır, gerçekleşmez ise ikinci boşlukta yer alan bloklar çalışır.



Proje 1) Ehliyet Alabilir misin?



g) Operatörler

Operatörler kategorisinde matematiksel işlemler, şart ifadelerin sonuçları, metin türünden ifadeler gibi birçok işlemi yaparken faydalanacağımız kod blokları mevcuttur.



Bloklar sırasıyla iki sayıyı toplamak, çıkarmak, çarpmak ve bölmek için kullanılır.



Belirtilen iki sayı değeri arasında rastgele bir sayı oluşturur.



İki ifade arasında soldaki ifadenin sağdaki ifadeden büyük olup olmadığını sorgular. Buraya girilecek değer sayısal ifadeler olabileceği gibi metin ifadeler de olabilir. $z > a$ ifadesi doğru (true) sonuç üretecektir. $e > k$ ifadesi ise (false) sonuç üretir.



İki ifade arasında soldaki ifadenin sağdaki ifadeden küçük olup olmadığını sorgular. Buraya girilecek değer sayısal ifadeler olabileceği gibi metin ifadeler de olabilir. $a < b$ ifadesi doğru (true) sonuç üretecektir. $c < b$



İki ifade arasında soldaki ifadenin sağdaki ifadeye eşit olup olmadığını sorgular. Buraya girilecek değer sayısal ifadeler olabileceği gibi metin ifadeler de olabilir.



VE operatörü iki şart ifadenin de doğru (true) sonuç üretmesini bekler. Her iki şart ifadede doğru(true) sonuç üretiyorsa VE operatörü de doğru sonuç üretir. Şart ifadelerden birisi yanlış(false) sonuç üretiyorsa VE operatörü de yanlış sonuç üretir.



VEYA operatörü kendisine verilen şart ifadelerden birisi dahi doğru sonuç üretmesi durumunda doğru(true) sonuç üretir. Şart ifadelerden her ikisi de yanlış sonuç üretiyorsa VEYA operatörü de yanlış (false) sonuç üretir.



Değil operatörü; bir şartın tersi değer üretir. Örneğin şart ifade olarak doğru(true) sonuç üreten bir ifade verilirse DEĞİL operatörü yanlış(false) sonuç üretecektir.

h) Değişkenler



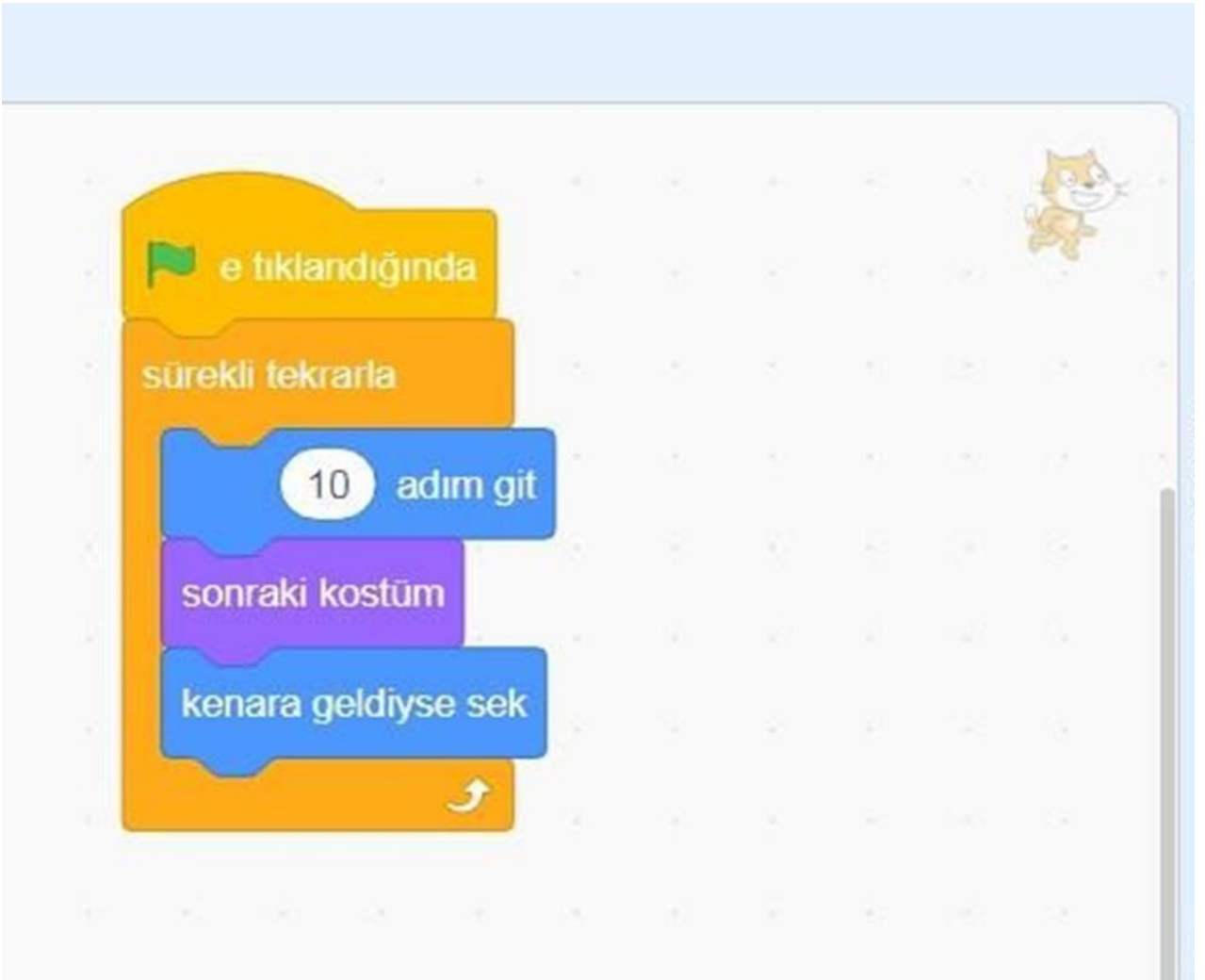
- Program, oyun sürecinde girilen sayı, yazı, doğru-yanlış, tarih vb. değerleri tutan programlama birimleridir.
- Değişkenleri biz tanımlarız ve isim veririz.
- Aynı isimde birden fazla değişken olamaz.

i) Sayaçlar



Yeşil bayrak tıklandığında program başlayarak her bir saniyede süre değişkenini 1 artırır. Bu sayede sayaç/kronometre programlanmış olur.

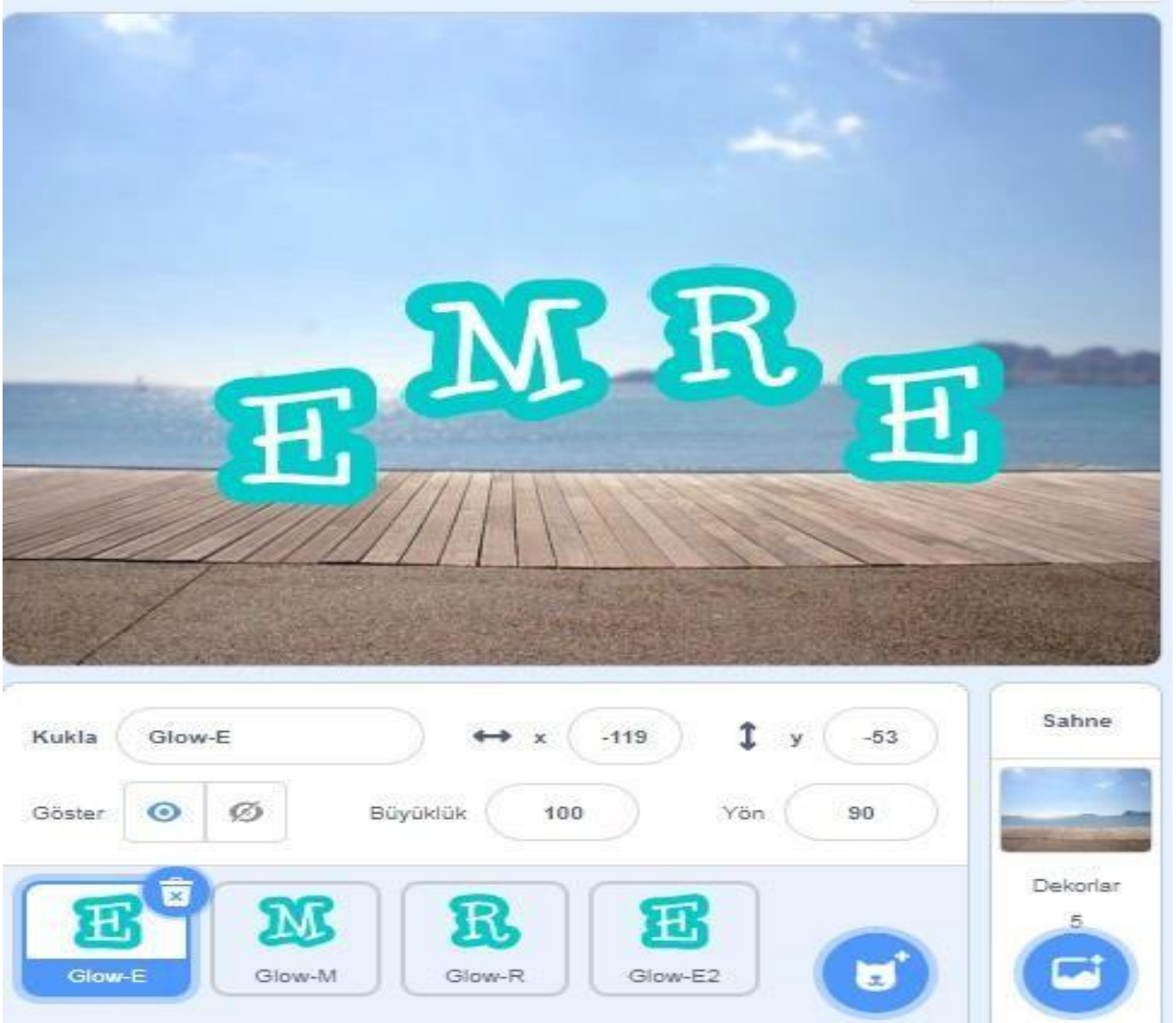
Proje 1) Kuklamızı Hareket Ettirme



Proje 2) Kuklaları Konuşturma



Proje 3) İsmimi Canlandır



ÖRNEK SORULAR

Aşağıdaki kelimeleri tanımlardaki uygun yerlere yerleştiriniz.

Problem

Akış Şeması

Yazılım

Algoritma

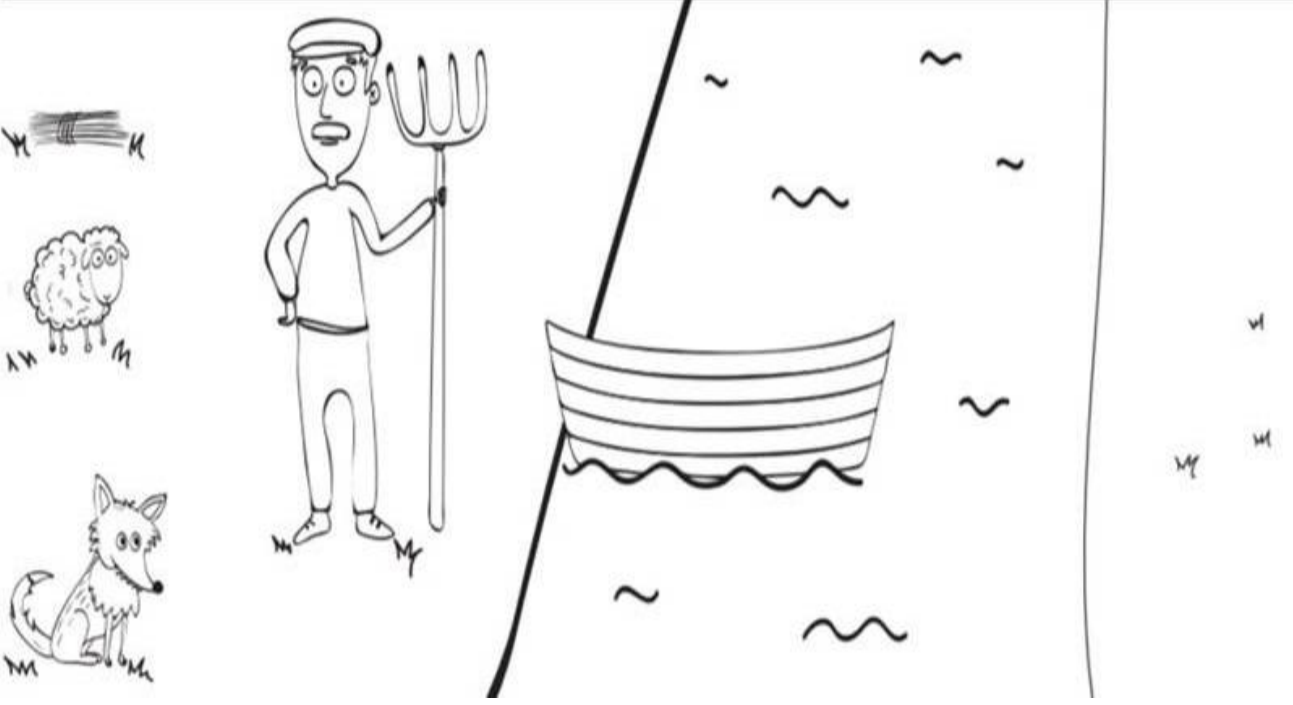
-: Çeşitli görevleri gerçekleştirmek amacıyla hazırlanmış programlara denir.
-: Çözülmesi gereken sorun yada aşılması gereken engeldir.
-: Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin adımlar halinde yazılması ile oluşur.
-: Bilgisayar programlarının işlem basamaklarının geometrik şekillerle gösteren şemadır.

Aşağıda yer alan 2 döngü bloğu arasındaki farkı yazınız.



KURT, KUZU VE OT

Ahmet Amca'nın çiftliği köyün biraz dışında Kızıldere'nin hemen öbür yanındaymış. Ahmet Amca bir gün kuzusunu, ormandan bahçesine inen kurdu ve kuzusu için ayırdığı bir miktar otu da alıp karşı kıyıya geçmek istemiş. Ancak karşıya geçebileceği tek araç ufacık bir kayıkmış ve hepsinin beraber karşıya geçmesi imkansızmış. Kayığa her defasında birini alabiliyormuş; ya kuzuyu ya kurdu ya da otu yanına alabilecekmış. Ancak bir sorunu daha varmış, kurtla kuzuyu yalnız bırakırsa kurt kuzuyu yermiş, kuzuyla otu yalnız bıraksa bu sefer kuzu da otları yermiş. Peki sizce nasıl olacak dya Ahmet Amca üçünü birden karşıya geçirecek?



Yukarıdaki problemin algoritmasını liste halinde yazınız.

1)

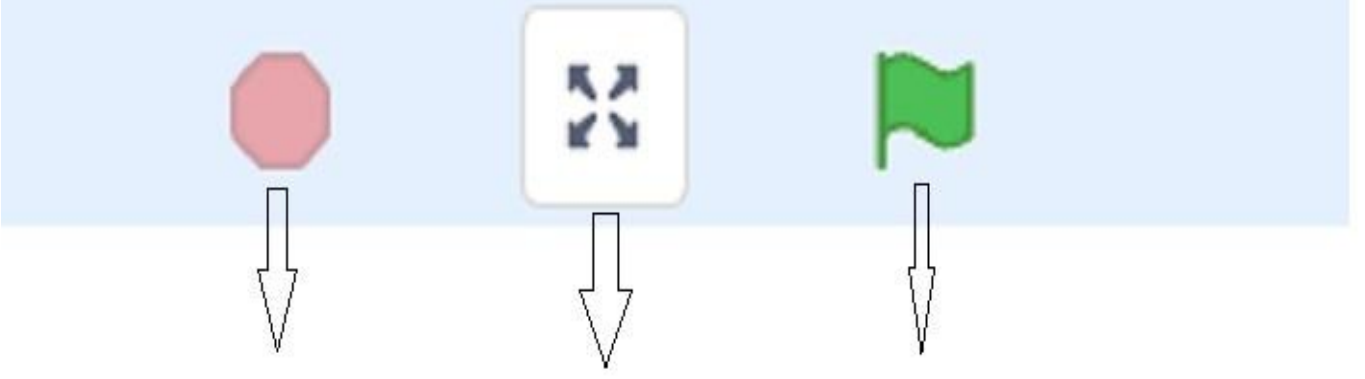
2)

3)

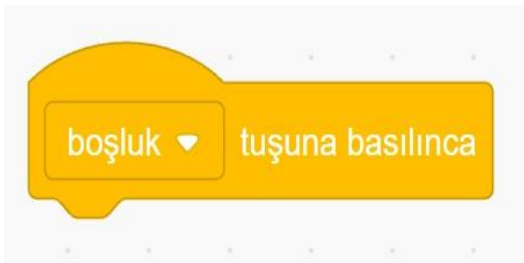
4)

5)

Aşağıda yer alan butonların görevlerini altlarına yazınız.



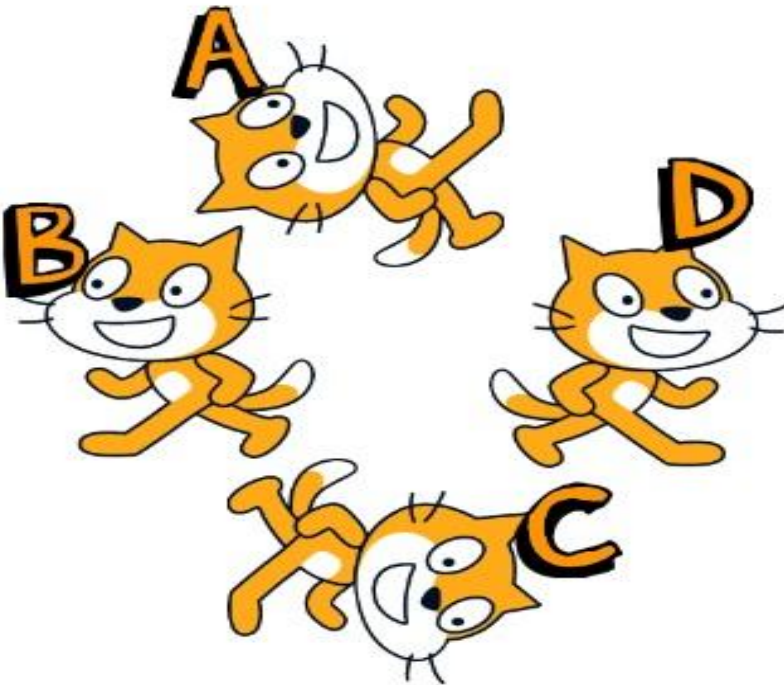
Aşağıda yer alan kod bloklarının görevlerini yanlarına yazınız.



Kedi karakterimizi kořuyormuř gibi gstermek iin ařađıda yer alan kod bloklarında soru iřaretli olan blgeye hangi blok gelmelidir?

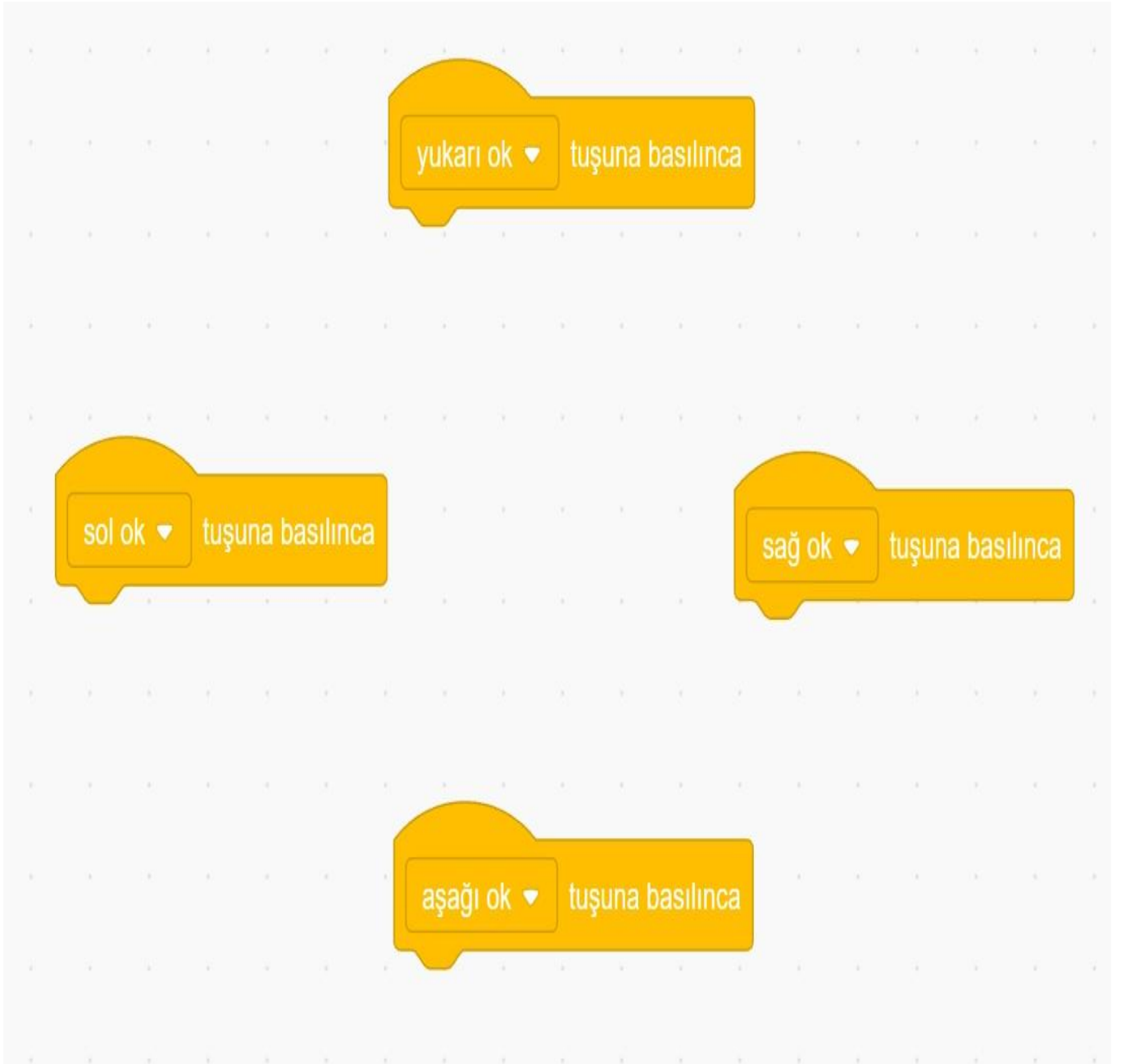


Ařađıda yer alan karakterlerin sahnede hangi yöne baktıklarını göstermek iin kullandığımız açılar yazınız.





Yukarıda yer alan blokları kullanarak, klavyede bulunan yön tuřları ile karakterimizi hareket ettirmek için gereken blokları uygun yerlerine yerleřtiriniz.



Aşağıda yer alan bloklar ayrı ayrı çalıştırıldığında karakterimiz hangi yöne hareket eder.

x konumunu 10 değiştir

y konumunu -10 değiştir

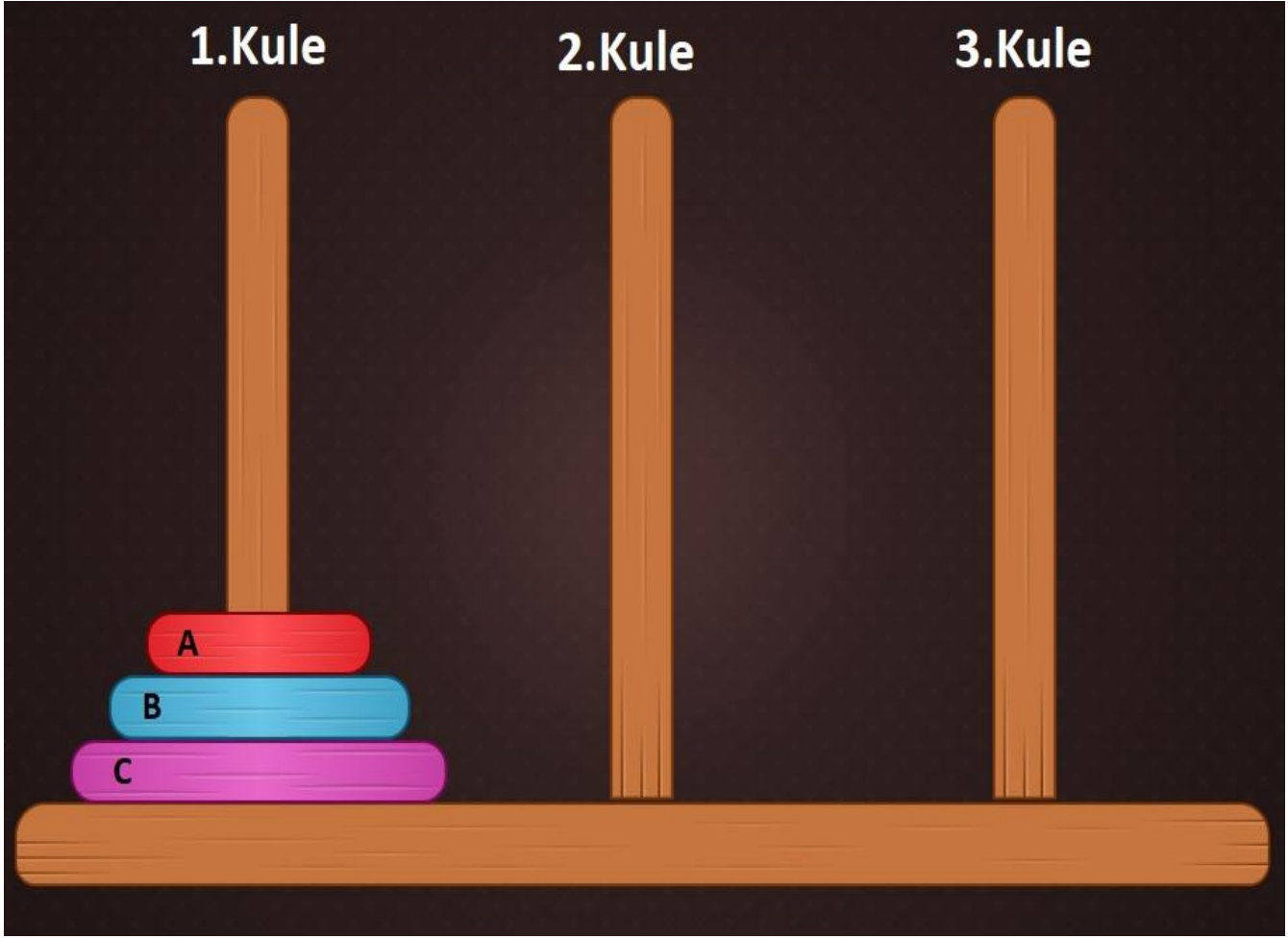
y konumunu 10 değiştir

x konumunu -10 değiştir

Aşağıda yer alan döngü bloklarının görevlerinin yazınız.



Aşağıda yer alan hanoi kuleleri probleminin çözüm algoritmasını liste halinde yazınız.



1)

2)

3)

4)

5)

6)

7)