

Bash kabuk programlamasında kod yazarken dikkat edilmesi gereken birkaç önemli husus vardır.

Önemli Hususlar ve Dikkat Edilmesi Gereken Noktalar

1. Yorum Satırları ve Kodun Anlaşılır Olması:

- Kodun anlaşılır ve okunabilir olması için yorum satırları ekleyin. Bu, kodunuzu başkalarının veya gelecekte kendinizin anlamasını kolaylaştırır.
- `#` ile başlayan satırlar yorum satırıdır ve betik çalıştırıldığında göz ardı edilir.

2. Boşluklar ve İndentasyon:

- Bash kabuk programında boşluklar önemli olabilir. Özellikle `if` ve `while` gibi kontrol yapılarında doğru yerlerde boşluk bırakmak, kodun doğru çalışmasını sağlar.
- `=` operatörü çevresinde boşluk bırakmayın (`var=value`). Ancak `[[...]]` içinde mantıksal ifadelerde boşluk bırakmalısınız (`[[$var == "value"]]`).

3. Değişken Atama ve Kullanım:

- Değişken atarken boşluk kullanmamak önemlidir: `var=value` doğru, `var = value` yanlıştır.
- Değişken kullanırken `$` işaretiyle değişkene erişilir: `echo $var`.

4. Komut Sonuçlarının Kontrolü:

- Komutların sonuçlarını kontrol etmek için `if` ve `elif` yapıları kullanın.
- `bc` ile hesaplama yaparken `scale=2` ifadesi, ondalık basamak sayısını belirtir.

5. Döngüler ve Kontrol Yapıları:

- Döngülerde ve `if` yapılarında doğru kapanış (`done`, `fi`) ve doğru mantıksal operatörler (`-lt`, `-gt`, `-eq`) kullanın.
- `case` yapısında her seçeneği `;` ile sonlandırmayı unutmayın.

6. Girdi Kontrolü ve Doğrulama:

- Kullanıcıdan alınan girdilerin geçerliliğini kontrol etmek önemlidir. Sayısal olmayan değerler veya boş girişler hatalara yol açabilir.
- Regex kullanarak (`[[...]]`) girdilerin doğruluğunu kontrol edin.

Örnek 1-

```
#!/bin/bash
```

```
# VKI hesaplama fonksiyonu
```

```
calculate_bmi() {
```

```
    local weight=$1
```

```
    local height=$2
```

```
    local bmi=$(echo "scale=2; $weight / (($height / 100) * ($height / 100))" | bc)
```

```
    echo $bmi
```

```
}
```

```
# VKI durumu belirleme fonksiyonu
```

```
determine_bmi_category() {
```

```
    local bmi=$1
```

```
    if (( $(echo "$bmi < 18.5" | bc -l) )); then
```

```
        echo "Zayıf"
```

```
    elif (( $(echo "$bmi >= 18.5 && $bmi < 24.9" | bc -l) )); then
```

```
        echo "Normal kilolu"
```

```
    elif (( $(echo "$bmi >= 25 && $bmi < 29.9" | bc -l) )); then
```

```
        echo "Fazla kilolu"
```

```
    else
```

```
        echo "Obez"
```

```
    fi
```

```
}
```

```
# Ana program
```

```
while true; do
```

```
    # Kullanıcıdan boy ve kilo bilgilerini al
```

```
    read -p "Boyunuzu cm olarak giriniz (örnek: 170): " height
```

```
    read -p "Kilonuzu kg olarak giriniz (örnek: 70): " weight
```

```
# Boy ve kilo değerlerinin sayısal olup olmadığını kontrol et
if ! [[ "$height" =~ ^[0-9]+(\.[0-9]+)?$ ]] || ! [[ "$weight" =~ ^[0-9]+(\.[0-9]+)?$ ]]; then
    echo "Lütfen geçerli sayısal değerler giriniz."
    continue
fi

# VKI hesapla
bmi=$(calculate_bmi $weight $height)
echo "Vücut Kitle İndeksiniz: $bmi"

# VKI durumunu belirle
category=$(determine_bmi_category $bmi)
echo "Durumunuz: $category"

# Kullanıcıya devam etmek isteyip istemediğini sor
read -p "Başka bir hesaplama yapmak ister misiniz? (E/H): " answer
case $answer in
    [Ee]* ) continue;;
    [Hh]* ) break;;
    * ) echo "Geçersiz giriş. Çıkılıyor."; break;;
esac
done

echo "Program sonlandırıldı."
```

Örnek-2

#!/bin/bash # Shebang, scriptin bir bash shell ile çalıştırılacağını belirtir.

VKI hesaplama fonksiyonu

```
calculate_bmi() {  
    local weight=$1 # Fonksiyona parametre olarak alınan ağırlık  
    local height=$2 # Fonksiyona parametre olarak alınan boy  
    local bmi=$(echo "scale=2; $weight / (($height / 100) * ($height / 100))" | bc) # bc kullanarak  
    ondalıklı hesaplama  
    echo $bmi # Hesaplanan VKI değerini döndür  
}
```

VKI durumu belirleme fonksiyonu

```
determine_bmi_category() {  
    local bmi=$1 # Fonksiyona parametre olarak alınan VKI değeri  
    if (( $(echo "$bmi < 18.5" | bc -l) )); then # Eğer VKI 18.5'ten küçükse  
        echo "Zayıf"  
    elif (( $(echo "$bmi >= 18.5 && $bmi < 24.9" | bc -l) )); then # Eğer VKI 18.5 ile 24.9 arasında ise  
        echo "Normal kilolu"  
    elif (( $(echo "$bmi >= 25 && $bmi < 29.9" | bc -l) )); then # Eğer VKI 25 ile 29.9 arasında ise  
        echo "Fazla kilolu"  
    else # Diğer durumlarda (VKI 30 ve üstü)  
        echo "Obez"  
    fi  
}
```

Ana program

```
while true; do # Sonsuz döngü başlatılır  
    # Kullanıcıdan boy ve kilo bilgilerini al  
    read -p "Boyunuzu cm olarak giriniz (örnek: 170): " height  
    read -p "Kilonuzu kg olarak giriniz (örnek: 70): " weight
```

```
# Boy ve kilo değerlerinin sayısal olup olmadığını kontrol et
if ! [[ "$height" =~ ^[0-9]+(\.[0-9]+)?$ ]] || ! [[ "$weight" =~ ^[0-9]+(\.[0-9]+)?$ ]]; then
    echo "Lütfen geçerli sayısal değerler giriniz."
    continue # Girdi geçersizse döngü başa döner
fi

# VKI hesapla
bmi=$(calculate_bmi $weight $height)
echo "Vücut Kitle İndeksiniz: $bmi"

# VKI durumunu belirle
category=$(determine_bmi_category $bmi)
echo "Durumunuz: $category"

# Kullanıcıya devam etmek isteyip istemediğini sor
read -p "Başka bir hesaplama yapmak ister misiniz? (E/H): " answer
case $answer in
    [Ee]* ) continue;; # 'E' veya 'e' girilirse döngü devam eder
    [Hh]* ) break;; # 'H' veya 'h' girilirse döngü kırılır ve program sonlanır
    * ) echo "Geçersiz giriş. Çıkılıyor."; break;; # Geçersiz giriş durumunda program sonlanır
esac
done

echo "Program sonlandırıldı."
```