



RUST

Vectorlerin Enum ile Kullanılması & Enum Çeşitlerinin Sayısal Olarak Kullanılması



Vector'de Çoklu Veri Tiplerinin Kullanılması

Vektörler sadece aynı türdeki verileri saklar

Bu bazen dezavantajdır.

Enum veri türleri kullanarak çoklu veri tiplerinde vektör oluşturabiliriz.



Vector'de Çoklu Veri Tiplerinin Kullanılması

Aşağıdaki gibi bir tablonun satırlarını vector ile saklayalım

Sıra no - integer	Boy -float	İsim - text
1	1.60	Fatma



Vector'de Çoklu Veri Tiplerinin Kullanılması

```
enum ExcelCalismaSayfasi {  
    SiraNo(i32),  
    Boy(f64),  
    Isim(String),  
}  
  
let row: Vec<ExcelCalismaSayfasi> = vec![  
    ExcelCalismaSayfasi::SiraNo(1),  
    ExcelCalismaSayfasi::Boy(1.56),  
    ExcelCalismaSayfasi::Isim(String::from("Fatma")),  
];  
  
println!("Enum isim : {:?})", row[0]);
```



```
let mut rows: Vec<Vec<ExcelCalismaSayfasi>> = Vec::new();
```

```
let row1: Vec<ExcelCalismaSayfasi> = vec![  
    ExcelCalismaSayfasi::SiraNo(2),  
    ExcelCalismaSayfasi::Boy(1.72),  
    ExcelCalismaSayfasi::Isim(String::from("Barış")),  
];
```

```
rows.push(row1);  
rows.push(row);  
  
println!("... {:?} ", rows);
```



Enum Çeşitlerinin Integer Olarak Döndürülmesi:

```
enum Season {  
    Spring,  
    Summer,  
    Autumn,  
    Winter,  
}
```

Enum çeşitleri, veri ile tanımlandıklarında, Spring(String) gibi, sayısal olarak döndürülemezler.

```
fn main() {  
    use Season::*;  
    let four_seasons = vec![Spring, Summer, Autumn, Winter];  
    for season in four_seasons {  
        println!("{}", season as u32);  
    }  
}
```




```
enum Star {  
    BrownDwarf = 10,  
    RedDwarf = 50,  
    YellowStar = 100,  
    RedGiant = 1000,  
    DeadStar, // Think about this one. What number will it have?  
}  
  
fn main() {  
    use Star::*;  
    let starvec = vec![BrownDwarf, RedDwarf, YellowStar, RedGiant];  
    for star in starvec {  
        match star as u32 {  
            size if size <= 80 => println!("Not the biggest star."), // Remember: size doesn't mean  
anything. It's just a name we chose so we can print it  
            size if size >= 80 => println!("This is a good-sized star."),  
            _ => println!("That star is pretty big!"),  
        }  
    }  
    println!("What about DeadStar? It's the number {}. ", DeadStar as u32);  
}
```





Celal AKSU

Bilişim Teknolojileri Öğretmeni

celalaksu@gmail.com

<https://www.linkedin.com/in/cllaksu/>

<https://twitter.com/ksacil>

<https://www.youtube.com/@eemcs>