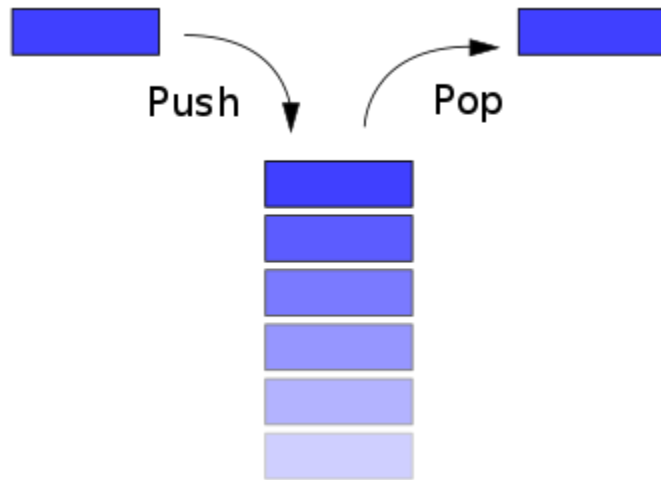


Stekovi i redovi

Stack



Generički stekovi Stack<T>

- Biblioteka **System.Collection.Generic**
- Klasa **Stack<T>**, gde je **T** tip podataka elemenata u steku
 - **Stack<int>** je stek celih brojeva
- Stack je **LIFO**(Last In, First Out) struktura
- Metoda **Push(T)** dodaje element na vrh steka
- Metoda **Pop()** uklanja i vraća element sa vrha steka(poslednji dodat element)
- Metoda **Peek()** vraća element sa vrha steka, ali ga ne uklanja
- Metoda **Clear()** uklanja sve elemente iz steka
- Svojstvo **Count** vraća broj elemenata u steku
- Metoda **Contains(T)** proverava da li se element nalazi u steku

Štampanje elemenata steka

```
static Stack<int> celobrojniStek = new Stack<int>();
```

```
static void Linija(int n)
{
    Console.WriteLine(new string('-', n));
}
```

```
static void Stampaj()
{
    if (celobrojniStek.Count == 0)
    {
        Console.WriteLine("Stek je prazan");
        return;
    }
    Linija(100);
    Console.Write("Vrh -> ");

    foreach (int i in celobrojniStek)
    {
        Console.Write($"{i} ");
    }
    Console.WriteLine("<- Dno");
    Linija(100);
}
```

Dodavanje novog elementa na vrh steka

```
static void DodajElement()
{
    while (true)
    {
        Console.Write("Unesite ceo broj: ");

        if (int.TryParse(Console.ReadLine(), out int temp))
        {
            celobrojniStek.Push(temp);
            Stampaj();
            return;
        }

        Console.WriteLine("Neispravan unos");
        Linija(100);
    }
}
```

Prikaz menija

```
static void PrikaziMeni()
{
    Console.WriteLine("Odaberite operaciju:");
    Console.WriteLine("1 -> dodaj element na vrh steka");
    Console.WriteLine("2 -> ukloni element sa vrha steka");
    Console.WriteLine("3 -> prikazi vrh steka");
    Console.WriteLine("4 -> izadji");
    Linija(100);
}
```

Main metoda

```
static void Main(string[] args)
{
    // Demonstracioni podaci
    celobrojniStek.Push(10);
    celobrojniStek.Push(20);
    celobrojniStek.Push(30);
    Stampaj();

    while (true)
    {
        PrikaziMeni();
        string operacija = Console.ReadLine();

        if (operacija == "4")
            break;

        switch (operacija)
        {
            case "1":
                DodajElement();
                break;
            // case 2:
            // case 3:
            default:
                Console.WriteLine("Nepodržana opcija");
                Linija(100);
                break;
        }
    }
}
```

Dodavanje novog elementa na vrh steka

```
C:\Users\goran\source\rep | x + v - □ x
-----
Vrh -> 30 20 10 <- Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
1
Unesite ceo broj: 23
-----
Vrh -> 23 30 20 10 <- Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
|
```

Uklanjanje elementa sa vrha steka

```
case "2":
    if (celobrojniStek.Count == 0)
    {
        Console.WriteLine("Stek je prazan");
        Linija(100);
        break;
    }

    int uklonjeni = celobrojniStek.Pop();
    Console.WriteLine($"Uklonjen element: {uklonjeni}");
    Stampaj();
    break;
```

Uklanjanje elementa sa vrha steka

```
C:\Users\goran\source\rep  X + v - □ X
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
2
Uklonjen element: 30
-----
Vrh -> 20 10 <- Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
|
```

Pregled vrha steka

```
case "3":  
    if (celobrojniStek.Count == 0)  
    {  
        Console.WriteLine("Stek je prazan");  
        Linija(100);  
        break;  
    }  
  
    Console.WriteLine("Vrh steka: " + celobrojniStek.Peek());  
    Linija(100);  
    break;
```

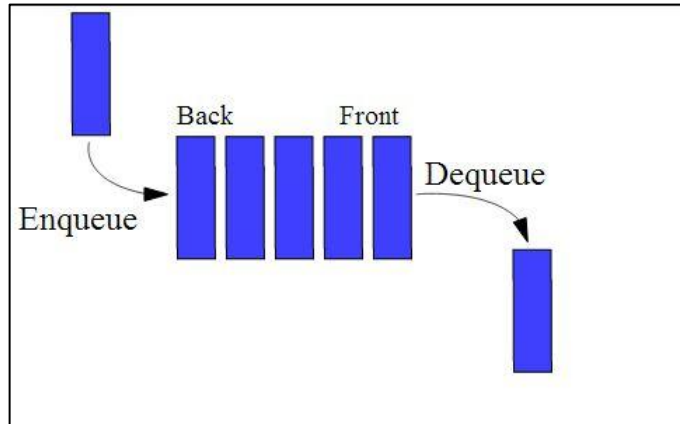
Pregled vrha steka

```
C:\Users\goran\source\rep | x + v - □ X
-----
Vrh -> 30 20 10 <- Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
3
Vrh steka: 30
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
-----
|
```

Izlazak iz aplikacije ili neispravan unos

```
else if (operacija == "4")
{
    break;
}
else
{
    Console.WriteLine("Neispravan unos");
    continue;
}
```

Red



Klasa Queue<T>

- **Queue<T>** predstavlja FIFO (First In, First Out) strukturu podataka – red
- Metoda **Enqueue(T)** dodaje element na kraj reda
- Metoda **Dequeue()** uklanja i vraća prvi element iz reda
- Svojstvo **Count** daje broj elemenata u redu
- Metoda **Contains(T)** proverava da li se element nalazi u redu
- Metoda **Peek()** vraća element sa početka reda, ali ga ne uklanja
- Metoda **Clear()** uklanja sve elemente iz reda

Štampanje reda

```
static Queue<int> celobrojniRed = new Queue<int>();
```

```
static void Stampaj()
{
    if (celobrojniRed.Count == 0)
    {
        Console.WriteLine("Red je prazan");
        return;
    }
    Linija(80);
    Console.Write("Prvi<-");
    foreach (int i in celobrojniRed)
    {
        Console.Write(i + " ");
    }
    Console.WriteLine("<-Poslednji");
    Linija(80);
}
```

Prikaz menija

```
static void PrikaziMeni()
{
    Console.WriteLine("Odaberite operaciju:");
    Console.WriteLine("1 -> dodaj element na kraj reda");
    Console.WriteLine("2 -> ukloni element sa pocetka reda");
    Console.WriteLine("3 -> prikazi pocetak reda");
    Console.WriteLine("4 -> izađi");
}
```

Dodavanje elementa na kraj reda

```
static void Dodaj()
{
    while (true)
    {
        Console.Write("Unesite ceo broj: ");
        if (int.TryParse(Console.ReadLine(), out int temp))
        {
            celobrojniRed.Enqueue(temp);
            Stampaj();
            break;
        }
        else
        {
            Console.WriteLine("Neispravan unos");
            Linija(100);
        }
    }
}
```

Main metoda

```
static void Main(string[] args)
{
    // Demonstracioni podaci
    celobrojniRed.Enqueue(10); celobrojniRed.Enqueue(20);
    celobrojniRed.Enqueue(30); celobrojniRed.Enqueue(50);
    Stampaj();
    while (true)
    {
        PrikaziMeni();
        string operacija = Console.ReadLine();
        switch (operacija)
        {
            case "1": Dodaj(); break;
            //2
            //3
            case "4": return;
            default:
                Console.WriteLine("Neispravan unos");
                Linija(100);
                break;
        }
    }
}
```

Dodavanje elementa na kraj reda

```
C:\Users\goran\source\rep... x + v - □ X

-----
Prvi<-10 20 30 50 <-Poslednji
-----

Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izardji
1
Unesite ceo broj: 32
-----

Prvi<-10 20 30 50 32 <-Poslednji
-----

Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izardji
|
```

Uklanjanje elementa sa početka reda

```
case "2":  
    if (celobrojniRed.Count == 0)  
    {  
        Console.WriteLine("Red je prazan");  
        break;  
    }  
    celobrojniRed.Dequeue();  
    Stampaj();  
    break;
```

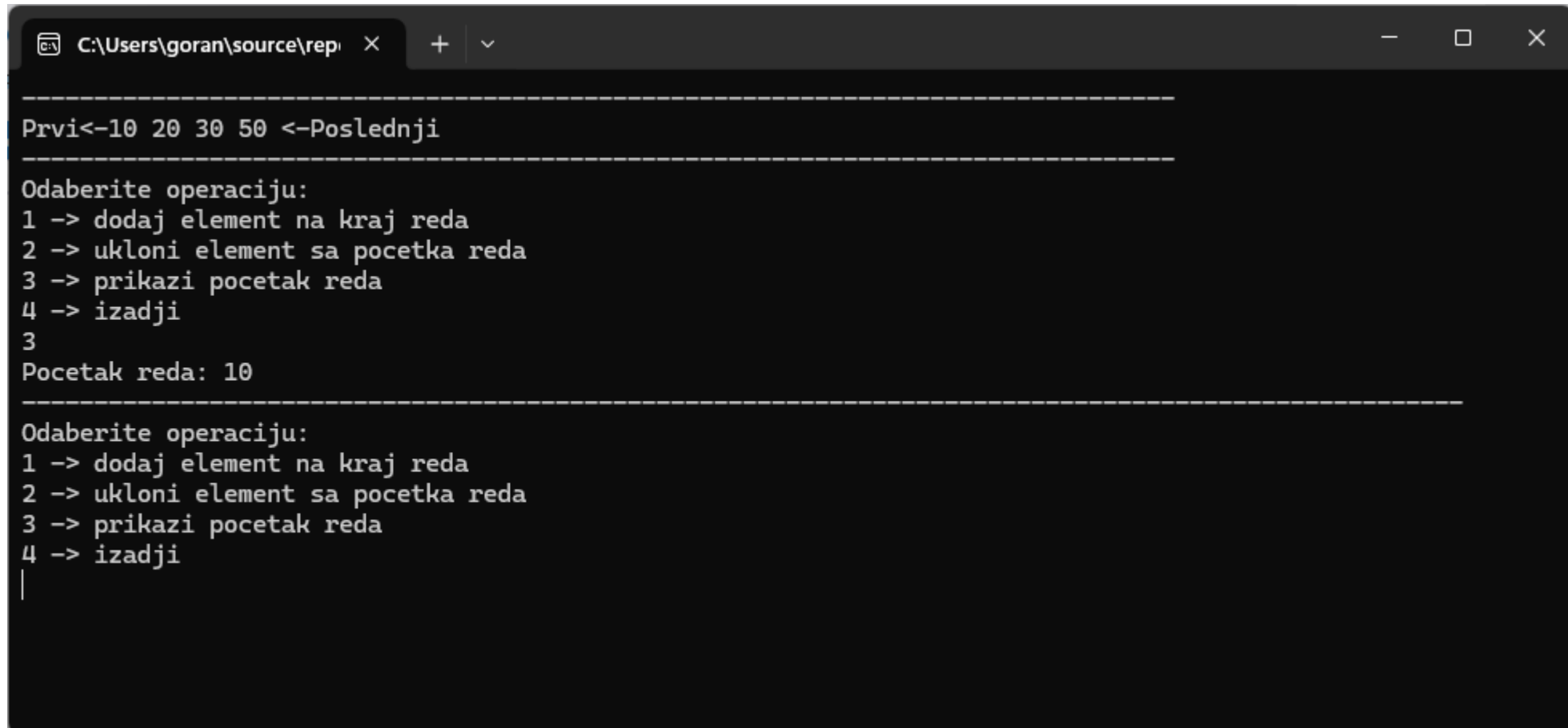
Uklanjanje elementa sa početka reda

```
C:\Users\goran\source\rep x + v
-----
Prvi<-10 20 30 50 <-Poslednji
-----
Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izardji
2
-----
Prvi<-20 30 50 <-Poslednji
-----
Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izardji
|
```

Prvi element u redu

```
case "3":  
    if (celobrojniRed.Count == 0)  
    {  
        Console.WriteLine("Red je prazan");  
        break;  
    }  
    Console.WriteLine("Pocetak reda: " + celobrojniRed.Peek());  
    Linija(100);  
    break;
```

Prvi element u redu



```
C:\Users\goran\source\rep> Prvi<-10 20 30 50 <-Poslednji

-----
Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izađi
3
Pocetak reda: 10
-----
Odaberite operaciju:
1 -> dodaj element na kraj reda
2 -> ukloni element sa pocetka reda
3 -> prikazi pocetak reda
4 -> izađi
|
```

Pitanje 1

Generički stek `Stack<T>` predstavlja:

- a. FIFO strukturu
- b. LIFO strukturu
- c. LILO strukturu

Odgovor: b

Pitanje 2

Stavljanje elementa **a** na vrh steka **st** vrši se korišćenjem naredbe:

- a. `st.Push(a);`
- b. `st.Pop(a);`
- c. `st.Push =a;`

Odgovor: a

Pitanje 3

Neka je definisan celobrojni red:

```
Queue<int> celobrojniRed = new Queue<int>();
```

Izvršavanjem naredbe:

```
celobrojniRed.Dequeue();
```

- a. dodaje se element na kraj reda
- b. uklanja se element sa kraja reda
- c. uklanja se element sa početka reda

Odgovor: c

Pitanje 4

Ako se u stek redom dodaju elementi 5, 10 i 15, redosled njihovog uklanjanja biće:

- a. 5, 10, 15
- b. 15, 10, 5
- c. 10, 5, 15

Odgovor:b

Pitanje 5

Ako se u red redom dodaju elementi 1, 2 i 3, koji element će biti prvi uklonjen?

- a. 1
- b. 3
- c. 2

Odgovor: a