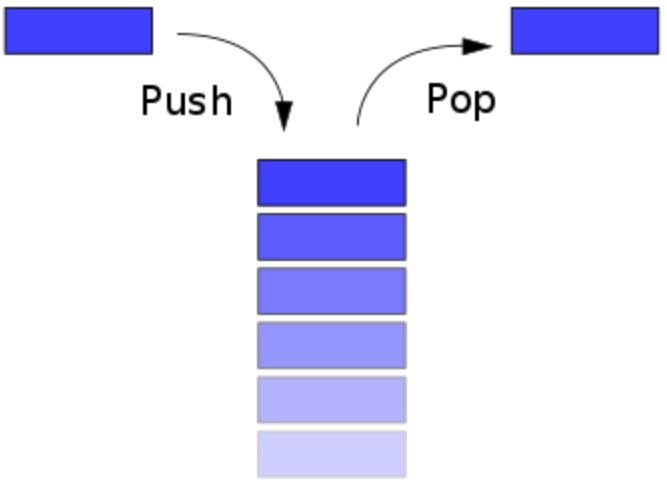


Stekovi i redovi

# Stack



# Generički stekovi Stack<T>

- Biblioteka System.Collection.Generics
- Klasa Stack<T>, gde je T tip podataka u steku
  - Stack<int> je stek celih brojeva
- Stack je LIFO(Last-in, first-out) struktura
- Metoda Push(T) stavlja objekat na stek
- Metoda Pop() skida objekat sa vrha steka (koji je poslednji stavljen na stek)
- Metoda Peek() vraća element koji je na vrhu steka ali ga ne uklanja
- Metoda Clear() uklanja sve objekte iz steka Stack<T>
- Svojstvo Count daje broj elemenata u steku
- Metoda Contains(T) utvrđuje da li se element nalazi u steku

# Štampanje elemenata steka

```
static void Linija(int n)
{
    Console.WriteLine("".PadRight(n, '_'));
}
```

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

```
static void Stampaj()
{
    if (celobrojniStek.Count > 0)
    {
        Linija(100);
        Console.Write("Vrh-->");
        foreach (int i in celobrojniStek)
        {
            Console.Write($" {i} |");
        }
        Console.WriteLine("--Dno");
        Linija(100);
    }
    else
    {
        Console.WriteLine("Stek je prazan");
    }
}
```

# Korisnički interfejs

```
static void Main(string[] args)
{
    celobrojniStek.Push(10);
    celobrojniStek.Push(20);
    celobrojniStek.Push(30);
    celobrojniStek.Push(50);
    Stampaj();
    string novaOperacija = "";
    string izbor;
    // do-while petlja
}
```

```
do
{
    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 -> izardji");
    Linija(100);
    izbor = Console.ReadLine();

    if (izbor == "1")
    {
        //1
    }
    else if (izbor == "2")
    {
        //2
    }
    else if (izbor == "3")
    {
        //3
    }
    else if (izbor == "4")
    {
        break;
    }
    else
    {
        Console.WriteLine("Neispravan unos");
        novaOperacija = "d";
        continue;
    }

    Console.WriteLine("Da li zelite novu operaciju? ");
    Console.WriteLine("Da -> Pritisni taster d");
    Console.WriteLine("Ne -> Pritisni taster n");
    Linija(100);

    novaOperacija = Console.ReadLine();
} while (novaOperacija == "d" || novaOperacija == "D");
```

# Stavljanje elementa na stek

```
//1
if (izbor == "1")
{
    Console.Write("Unesite ceo broj: ");
    if (int.TryParse(Console.ReadLine(), out int temp))
    {
        celobrojniStek.Push(temp);
        Stampaj();
    }
    else
    {
        Console.WriteLine("Neispravan unos");
        Linija(100);
    }
}
}
```

# Stavljanje elementa na stek

```
C:\Users\goran\source\repos\ConsoleASp10\ConsoleASp10\bin\Debug\ConsoleASp10.exe
Vrh--> 50 | 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: 55
Vrh--> 55 | 50 | 30 | 20 | 10 |--Dno
Da li zelite novu operaciju?
Da -> Pritisni taster d
Ne -> Pritisni taster n
-
```

# Skidanje elementa sa steka

```
//2
if (celobrojniStek.Count >0)
{
    celobrojniStek.Pop();
    Stampaj();
}
else
{
    Console.WriteLine("Stek je prazan");
}
```

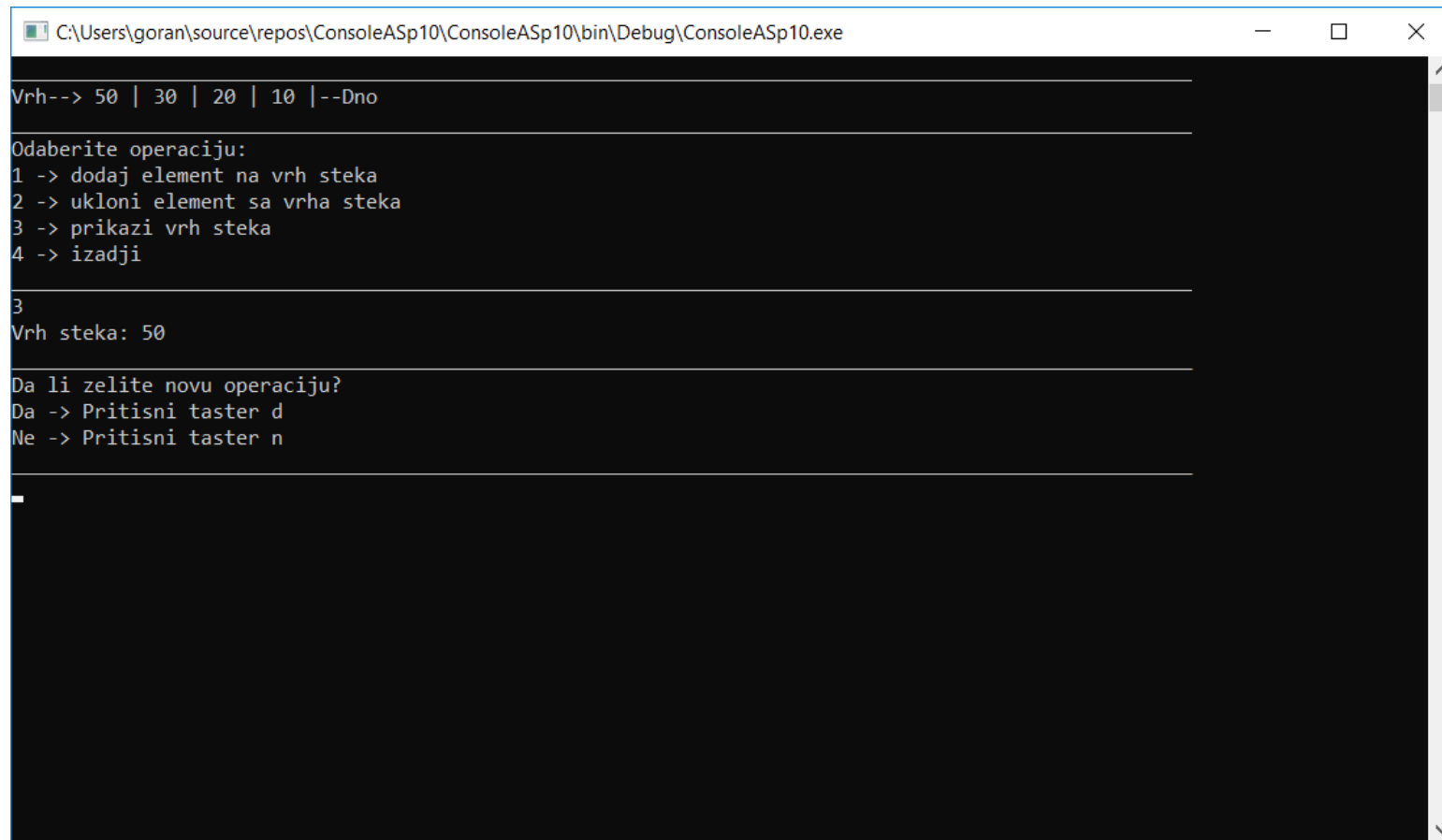


# Skidanje elementa sa steka

```
C:\Users\goran\source\repos\ConsoleASp10\ConsoleASp10\bin\Debug\ConsoleASp10.exe
Vrh--> 50 | 30 | 20 | 10 |--Dno
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izađi
2
Vrh--> 30 | 20 | 10 |--Dno
Da li zelite novu operaciju?
Da -> Pritisni taster d
Ne -> Pritisni taster n
```

# Pregled vrha steka

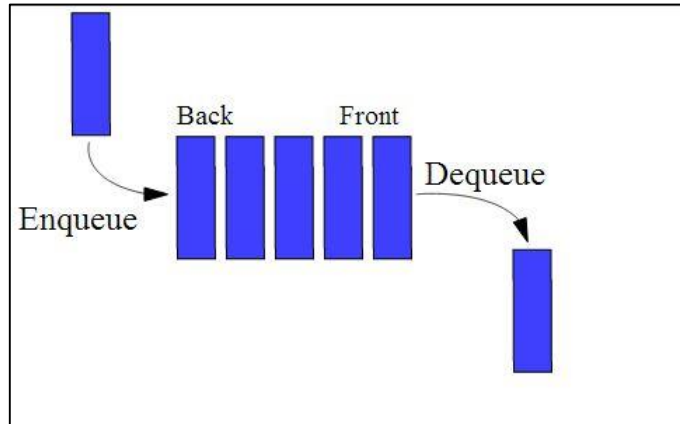
```
//3  
Console.WriteLine("Vrh steka: " + celobrojniStek.Peek());  
Linija(100);
```



The screenshot shows a console window titled "C:\Users\goran\source\repos\ConsoleASp10\ConsoleASp10\bin\Debug\ConsoleASp10.exe". The output in the console is as follows:

```
Vrh--> 50 | 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: 50  
  
Da li zelite novu operaciju?  
Da -> Pritisni taster d  
Ne -> Pritisni taster n
```

# Red



# Klasa Queue<T>

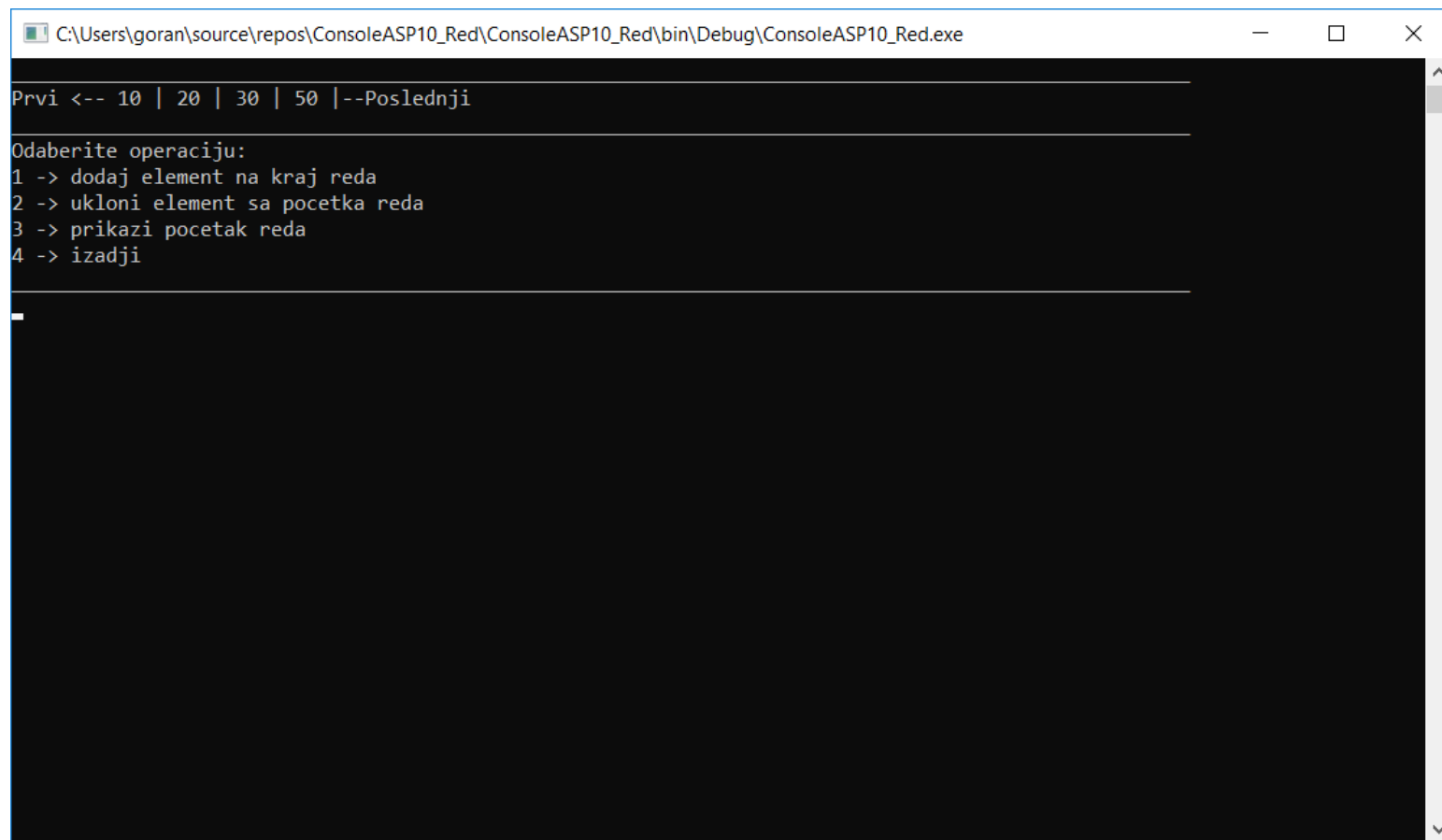
- Queue<T> predstavlja FIFO kolekciju objekata (first –in, first-out)
- Metoda Enqueue(T) dodaje objekat na kraj reda
- Metoda Dequeue() uklanja prvi element iz reda
- Svojstvo Count daje broj elemenata uredu
- Metoda Contains(T) utvrđuje da li se element nalazi u redu
- Metoda Peek() vraća objekat sa početka reda bez njegovog uklanjanja iz reda
- Metoda Clear() uklanja sve objekte iz reda Queue<T>

# Štampanje reda

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

```
static void Stampaj()
{
    if (celobrojniRed.Count > 0)
    {
        Linija(100);
        Console.Write("Prvi <--");
        foreach (int i in celobrojniRed)
        {
            Console.Write($" {i} |");
        }
        Console.WriteLine("--Poslednji");
        Linija(100);
    }
    else
    {
        Console.WriteLine("Red je prazan");
    }
}
```

# Korisnički interfejs



```
C:\Users\goran\source\repos\ConsoleASP10_Red\ConsoleASP10_Red\bin\Debug\ConsoleASP10_Red.exe
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
```

# Ubacivanje u red

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

# Ubacivanje u red

```
C:\Users\goran\source\repos\ConsoleASP10_Red\ConsoleASP10_Red\bin\Debug\ConsoleASP10_Red.exe
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
1
Unesite ceo broj: 80
Prvi <-- 10 | 20 | 30 | 50 | 80 |--Poslednji
Da li zelite novu operaciju?
Da -> Pritisni taster d
Ne -> Pritisni taster n
-
```



# Prvi element u redu

```
Console.WriteLine("Pocetak reda: " + celobrojniRed.Peek());  
Linija(100);
```



```
C:\Users\goran\source\repos\ConsoleASP10_Red\ConsoleASP10_Red\bin\Debug\ConsoleASP10_Red.exe  
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  
Da li zelite novu operaciju?  
Da -> Pritisni taster d  
Ne -> Pritisni taster n  
-
```

# Uklanjanje elementa sa početka reda

```
if (celobrojniRed.Count > 0)
{
    celobrojniRed.Dequeue();
    Stampaj();
}
else
{
    Console.WriteLine("Stek je prazan");
}
```

# Uklanjanje elementa sa početka reda



```
C:\Users\goran\source\repos\ConsoleASP10_Red\ConsoleASP10_Red\bin\Debug\ConsoleASP10_Red.exe

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

2

Prvi <-- 20 | 30 | 50 |--Poslednji

Da li zelite novu operaciju?
Da -> Pritisni taster d
Ne -> Pritisni taster n
```

# Pitanje 1

Generički stek `Stack<T>` kada se instancira 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:

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

Linijom koda:

```
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