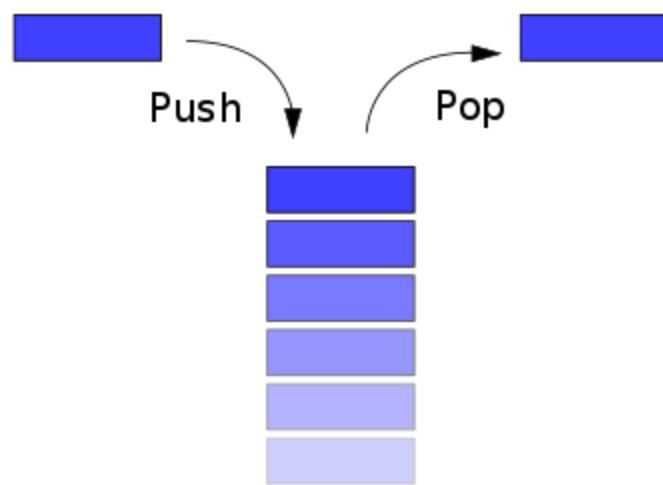


Stekovi i redovi

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

```
ConsoleAsp10_Stack
Program.cs*  X
ConsoleAsp10_Stack  2  ConsoleAsp10_Stack.Program  Stampaj()
0 references

55     static void Main(string[] args)
56     {
57         celobrojniStek.Push(10);
58         celobrojniStek.Push(20);
59         celobrojniStek.Push(30);
60         celobrojniStek.Push(50);
61         Stampaj();
62         string operacija;
63         while (true)
64         {
65             Console.WriteLine("Odaberite operaciju:");
66             Console.WriteLine("1 -> dodaj element na vrh steka");
67             Console.WriteLine("2 -> ukloni element sa vrha steka");
68             Console.WriteLine("3 -> prikazi vrh steka");
69             Console.WriteLine("4 -> izadjи");
70             Linija(100);
71             operacija = Console.ReadLine();
72
73             if (operacija == "1")...
74             else if (operacija == "2")...
75             else if (operacija == "3")...
76             else if (operacija == "4")...
77             else...
78         }
79     }
80
81     static void Stampaj()
82     {
83         for (int i = 0; i < celobrojniStek.Count; i++)
84         {
85             Console.WriteLine(celobrojniStek[i]);
86         }
87     }
88
89     static void Linija(int broj)
90     {
91         for (int i = 0; i < broj; i++)
92         {
93             Console.Write("-");
94         }
95         Console.WriteLine();
96     }
97
98
99
100
101
102
103
104
105 }
```

The code in `Program.cs` defines a `Main` method that initializes a stack (`celobrojniStek`) with four elements (10, 20, 30, 50) and prints them. It then enters a loop where it prompts the user for an operation (1-4). The operations correspond to pushing an element onto the stack, popping an element from the stack, peeking at the top element, and exiting the program respectively. The `Stampaj()` method is used to print the current state of the stack, and `Linija()` is used to print a horizontal line.

Stavljanje elementa na stek

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

```
if (operacija == "1")
{
    DodajElement();
}
```

Stavljanje elementa na stek

```
C:\Users\goran\source\rep... X + | - □ ×

-----
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 -> izadji
-----
1
Unesite ceo broj: 23
-----
Vrh--> 23 | 50 | 30 | 20 | 10 |--Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izadji
-----|-----
```

Skidanje elementa sa steka

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

Skidanje elementa sa steka

```
C:\Users\goran\source\rep... X + | - □ ×

1
Unesite ceo broj: 23
-----
Vrh--> 23 | 50 | 30 | 20 | 10 |--Dno
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izadji
-----
2
-----
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 -> izadji
-----
```

Pregled vrha steka

```
else if (operacija == "3")
{
    Console.WriteLine("Vrh steka: " + celobrojniStek.Peek());
    Linija(100);
}
```

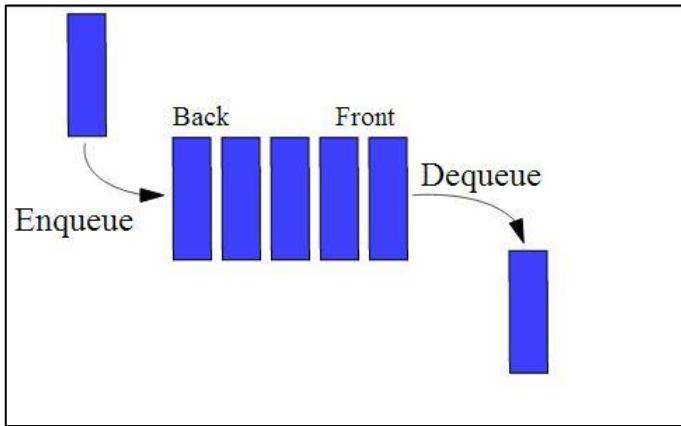
Pregled vrha steka

```
C:\Users\goran\source\rep... + - X
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 -> izadji
-----
3
Vrh steka: 50
-----
Odaberite operaciju:
1 -> dodaj element na vrh steka
2 -> ukloni element sa vrha steka
3 -> prikazi vrh steka
4 -> izadji
-----
```

Izlazak iz aplikacije ili neispravan unos

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

Red



Klasa Queue<T>

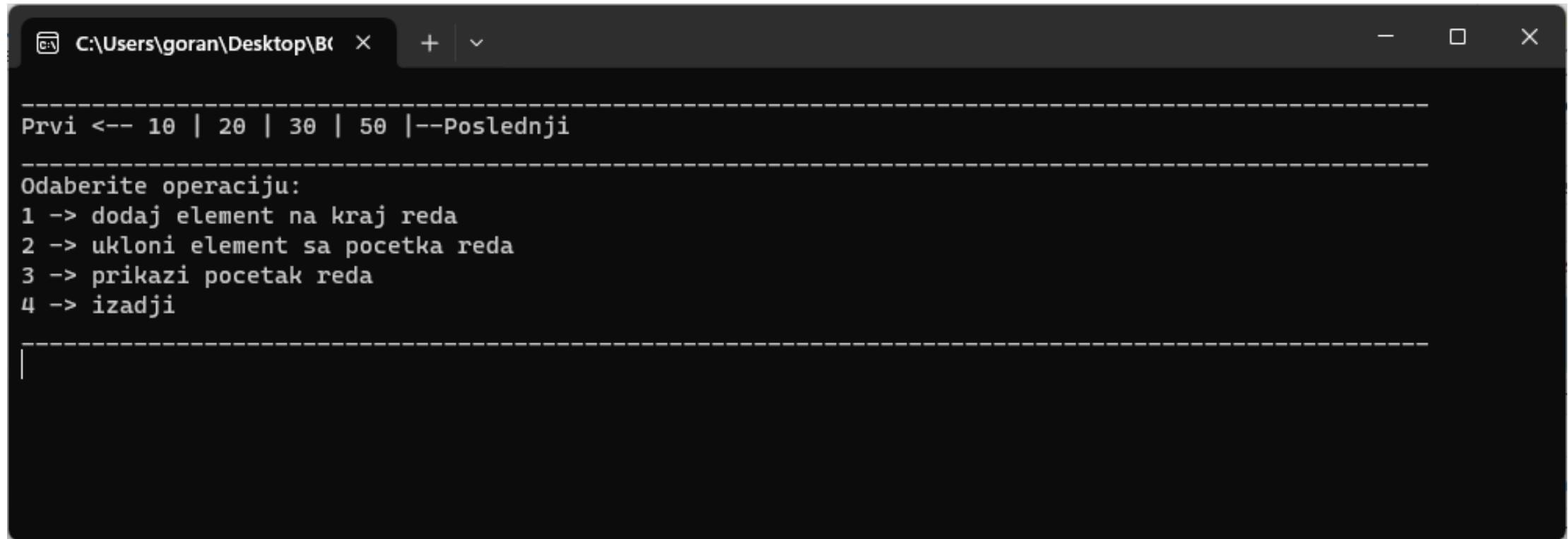
- Queue<T> predstavlja FIFO kolekciju objekata (first –in, first-out)
- Metoda Enqueue(T) dodaje objekat na kraj reda
- Metoda Dequeue() uklanja prvi element iz reda
- Svojstvo Count daje broj elemenata uredu
- Metoda Contains(T) utvrđuje da li se element nalazi u redu
- Metoda Peek() vraća objekat sa 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.WriteLine("Prvi ---");
        foreach (int i in celobrojniRed)
        {
            Console.Write($" {i} | ");
        }
        Console.WriteLine("--Poslednji");
        Linija(100);
    }
    else
    {
        Console.WriteLine("Red je prazan");
    }
}
```

Korisnički interfejs



The screenshot shows a terminal window with the following content:

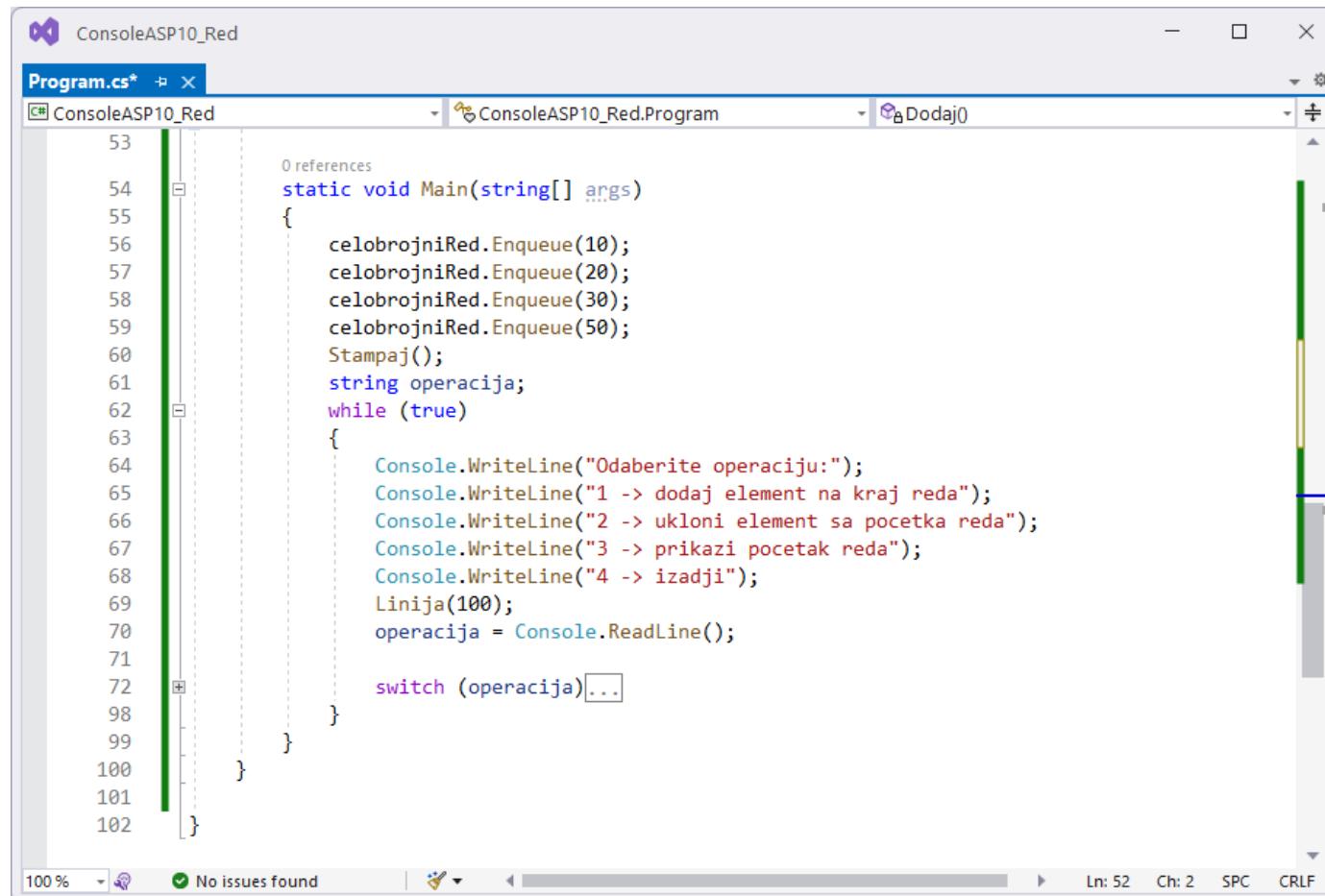
```
C:\Users\goran\Desktop\B... + | - □ ×
```

```
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 -> izadji
```

A cursor is visible at the bottom of the terminal window.

Realizacija interfejsa



```
ConsoleASP10_Red
Program.cs*  X
ConsoleASP10_Red  ConsoleASP10_Red.Program  Dodaj()
53
54     static void Main(string[] args)
55     {
56         celobrojniRed.Enqueue(10);
57         celobrojniRed.Enqueue(20);
58         celobrojniRed.Enqueue(30);
59         celobrojniRed.Enqueue(50);
60         Stampaj();
61         string operacija;
62         while (true)
63         {
64             Console.WriteLine("Odaberite operaciju:");
65             Console.WriteLine("1 -> dodaj element na kraj reda");
66             Console.WriteLine("2 -> ukloni element sa pocetka reda");
67             Console.WriteLine("3 -> prikazi pocetak reda");
68             Console.WriteLine("4 -> izadjи");
69             Linija(100);
70             operacija = Console.ReadLine();
71
72             switch (operacija)
73             {
74                 case "1":
75                     int noviElement;
76                     Console.WriteLine("Unesite novi element: ");
77                     noviElement = int.Parse(Console.ReadLine());
78                     celobrojniRed.Enqueue(noviElement);
79                     break;
80
81                 case "2":
82                     celobrojniRed.Dequeue();
83                     break;
84
85                 case "3":
86                     celobrojniRed.Print();
87                     break;
88
89                 case "4":
90                     return;
91
92                 default:
93                     Console.WriteLine("Neispravna operacija!");
94                     break;
95             }
96         }
97     }
98
99 }
100
101
102 }
```

Ubacivanje u red

```
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);
        }
    }
}
```

```
case "1":
    Dodaj();
    break;
```

Ubacivanje u red

```
C:\Users\goran\Desktop\B C: \Users\goran\Desktop\B 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 -> izadjи

1
Unesite ceo broj: 35

Prvi <-- 10 | 20 | 30 | 50 | 35 |--Poslednji

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

|
```

Uklanjanje elementa sa početka reda

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

Uklanjanje elementa sa početka reda

```
C:\Users\goran\Desktop\B> + ▾
-----
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 -> izadji
-----
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 -> izadji
-----|-----
```

Prvi element u redu

```
case "3":  
    Console.WriteLine("Pocetak reda: " + celobrojniRed.Peek());  
    Linija(100);  
    break;
```

Prvi element u redu

```
C:\Users\goran\Desktop\B C:\Users\goran\Desktop\B + - 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 -> izadji  
-----  
3  
Pocetak reda: 10  
-----  
Odaberite operaciju:  
1 -> dodaj element na kraj reda  
2 -> ukloni element sa pocetka reda  
3 -> prikazi pocetak reda  
4 -> izadji  
-----|
```

Izlazak iz aplikacije ili neispravan unos

```
case "4":  
    return;  
default:  
    Console.WriteLine("Neispravan unos");  
    break;
```

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