Universiteti i Prishtinës

Fakulteti i Inxhinierisë Elektrike dhe Kompjuterike

Algoritmet dhe struktura e të dhënave

Vehbi Neziri

FIEK, Prishtinë 2015/2016

Java 7

- Abstract Data Type (ADT)
- Steku
- Queue
- STL Steku
- STL Queue

ADT

- Abstract Data Type (ADT) - model matematik i objekteve të të dhënave dhe funksionet që veprojnë në këto objekte.
- ADT të përbëhet nga dy pjesë: **objektet e të dhënave** dhe operacionet me objektet e të dhënave.
- Termi ***data type*** i referohet implementimit të modelit matematik të specifikuar nga një ADT.
- Termi ***data structure*** i referohet koleksionit të variablave që janë të lidhura në ndonjë mënyrë të veçantë.
- Nocioni ***data type*** përfshin ***basic data types***.

ADT

- Abstract Data Type ka tri gjëra të cilat e shoqërojnë, si:
  - Emrin e ADT
  - Grupin e vlerave që përkojnë në ADT – domeni
  - Grupin e operacioneve në të dhëna.

Shembull 7.1

- Duke u bazuar në veçoritë e ADT-së të definohen emri, grupi i vlerave dhe grupi i operacioneve për ORËN.

emriTipiDhenave

tipiORE

domeni

secili tipiORE është koha e ditës në formatin e orëve, minutave dhe sekondave

Operacionet

vendos kohën

kthe kohën

printo kohën

rrit kohën për një sekondë

rrit kohën për një minutë

rrit kohën për një orë

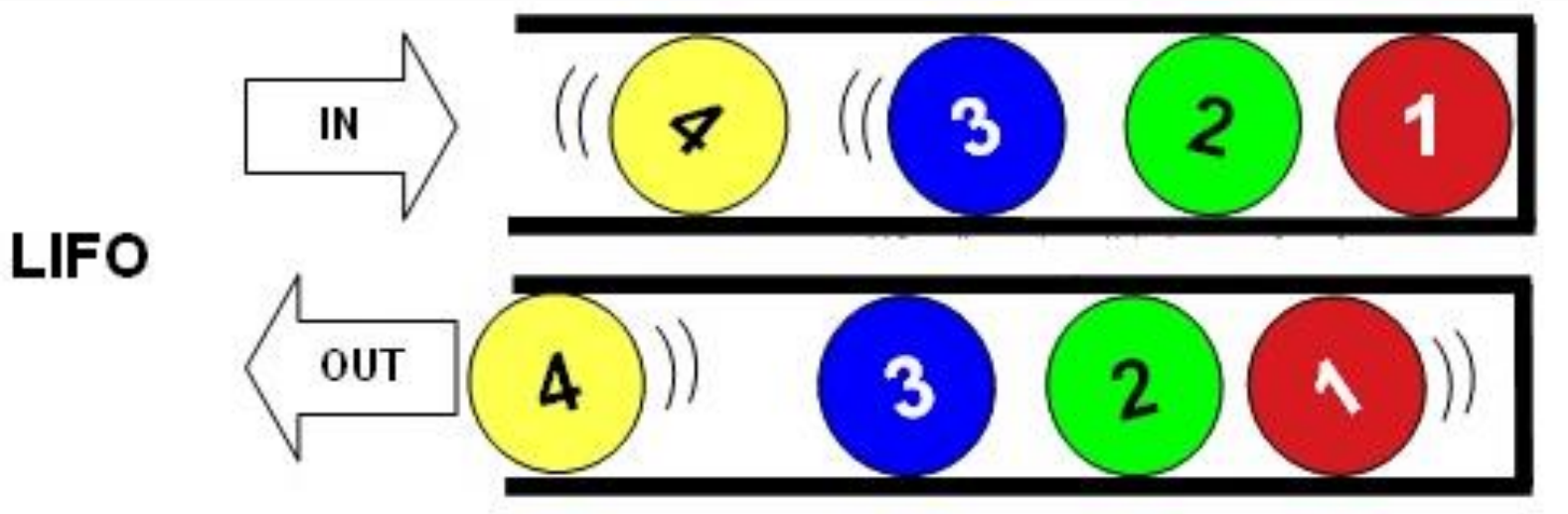
Shembull 7.2

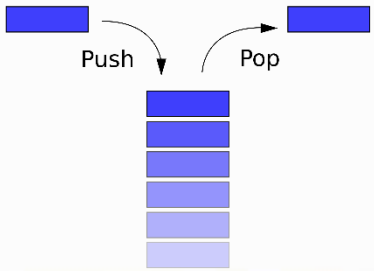
- Shembulli paraprak të shkruhet në C++. (Të implenetohet vetëm klasa me prototipat e funksioneve)

```
1  #include <iostream>
2  using namespace std;
3
4  class tipiORE
5  {
6  public:
7      void vendosKohen(int ora, int minuta, int sekonda);
8      void ktheKohen(int& ora, int& minuta, int& sekonda);
9      void printoKohen();
10     void rritSekondat();
11     void rritMinutat();
12     void rritOren();
13
14     tipiORE(int ora, int minuta, int sekonda);
15     tipiORE();
16 private:
17     int ora;
18     int min;
19     int sek;
20 };
```

Steku

- **Steku** – është formë e veçantë e grumbullit me semantikën LIFO (Last In-First Out)





Steku (2)

- **ADT** mbulon një sërë të objekteve si dhe operacionet e kryera në këto objekte:
  - Initialize(S)
  - Push(x)
  - Pop
  - Top
  - Kill(S)

Shembull 7.3

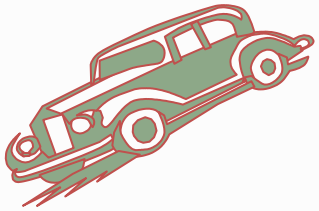
- Të shkruhet programi në të cilin deklarohet klasa **stack** dhe funksionet **push()** dhe **pop()** për futje, respektivisht për heqje të objekteve si dhe funksioni **paraqit** për shtypje të përmbajtjes së stekut.

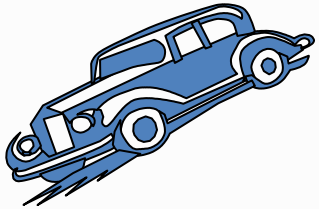
```cpp
#include <iostream>
using namespace std;
#define SIZE 3

class stack
{
private:
    int vargu[SIZE];
    int pozita;
public:
    stack();
    void push(int a); //Shto ne stek
    int pop();        //Terhiq/Fshi nga steku.
    void paraqit();
};

stack::stack()
{
    pozita = -1;      //Për të treguar stekun e zbrazet
}

void stack::push(int a)
{
    pozita++;
    if (pozita<SIZE)
    {
        vargu[pozita] = a;
    }
    else
    {
        cout << "Steku eshte i mbushur!!" << endl;
        pozita--;
    }
}

int stack::pop()
{
    if (pozita == -1){
        cout << "Steku eshte i zbrazet!!!" << endl;
        return NULL;
    }
    else{
        int vlera = vargu[pozita];  // Vleren e pozites ruaje ne vlera
        vargu[pozita] = NULL;       // Beje lokacionin origjinal ne NULL
        pozita--;                  // Dekrementoje poziten per 1
        return vlera;             // Kthe elementin e fshirë
    }
}
void stack::paraqit()
{
    int t = pozita;
    cout << "\nVlerat aktuale ne stek jane:\n";
    while (t >= 0)
    {
        cout << vargu[t] << endl;
        t--;
    }
}
int main()
{
    stack a;
    a.push(1);
    cout << "1 u shtye (Push) ne stek\n";
    a.push(2);
    cout << "2 u shtye (Push) ne stek\n";
    a.push(3);
    cout << "3 u shtye (Push) ne stek\n\n";

    a.paraqit();
    cout << a.pop() << " u nxorr (Pop)\n";
    cout << a.pop() << " u nxorr (Pop)\n";
    cout << a.pop() << " u nxorr (Pop)\n";
    a.paraqit();
    return 0;
}
```

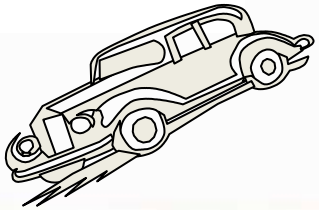
Queue

- **Queue** – është kontejner në të cilin:
  - Ndërfutja bëhet vetëm nga fundi
  - Fshirjet dhe modifikimet bëhen vetëm nga kreu

FIFO



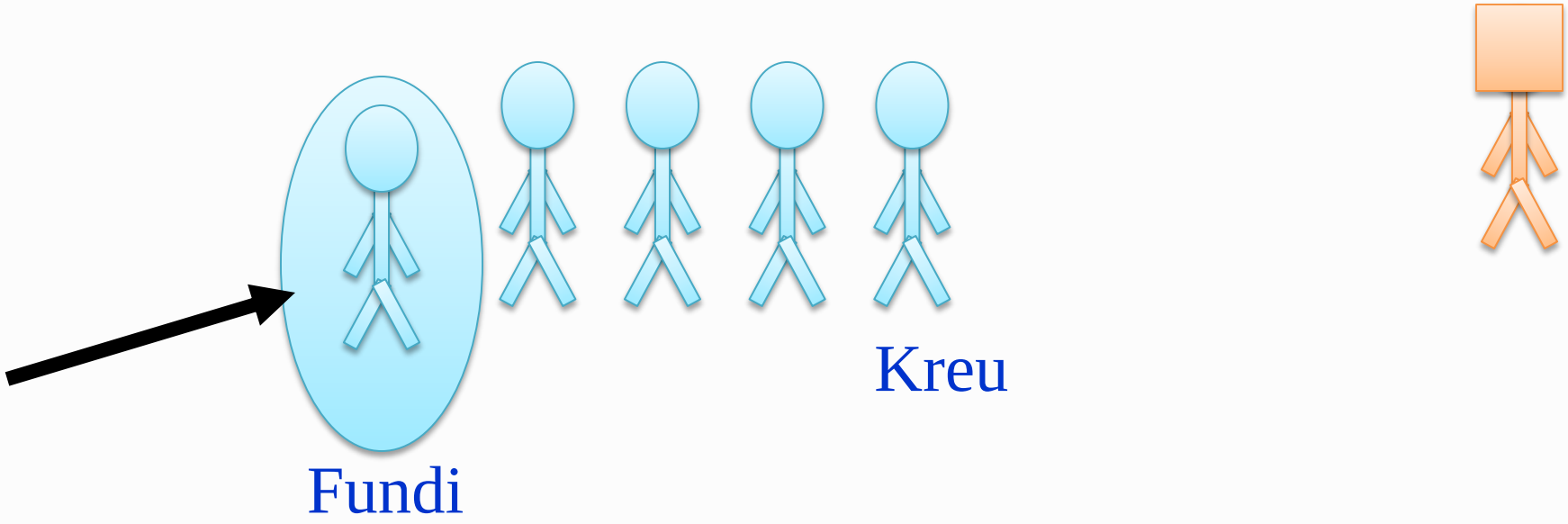




Veprimet në queue

- **Queue** – është si një rresht i njerëzve që ka kreun dhe fundin

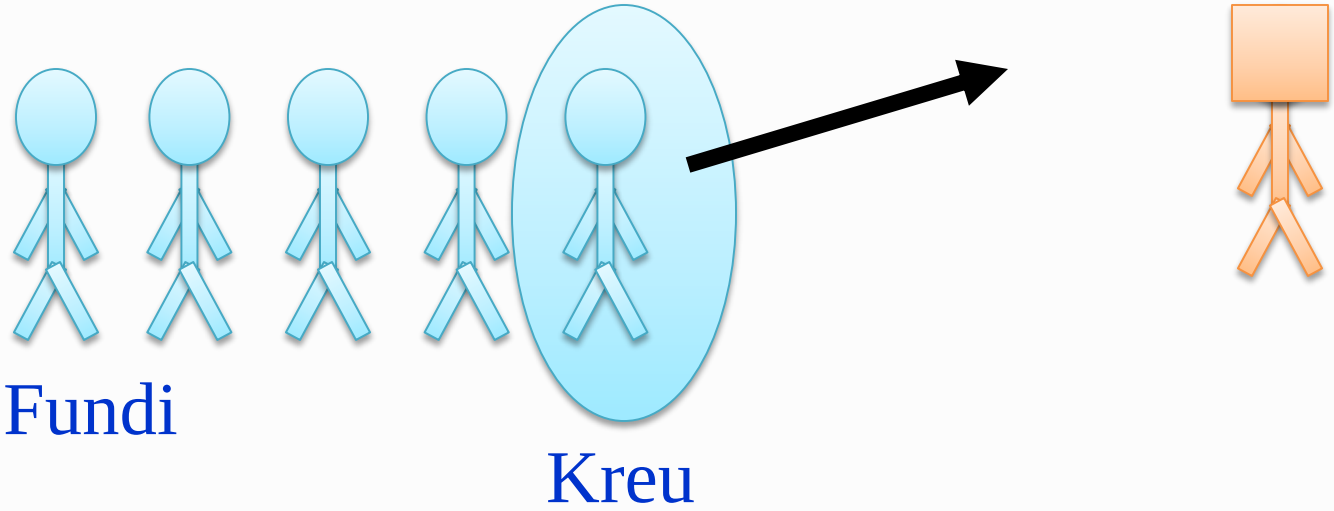
- Njerëzit e rinj mund të futen në radhë vetëm në fund. Kjo në C++ quhet ***push***, operacion ***enqueue***



Veprimet në queue

- Kur një objekt merret nga radha, ai çdo herë thirret nga fillimi/kreu.

- Në C++ kjo quhet ***pop***, operacion ***dequeue***



Shembull 7.4

- Të shkruhet programi në të cilin implementohet klasa **queue** dhe funksionet **nderfut, largo** dhe **shfaq** për mbushje, largim dhe shfaqje të përmbajtjes së queue.

```cpp
#include<iostream>
using namespace std;
#define size 4

class queue
{
    int radha[size];
    int kreu, fundi;
public:
    queue()
    {
        kreu = -1; fundi = -1;
    }
    void nderfut();
    void largo();
    void shfaq();
};

void queue::nderfut()
{
    if (fundi == size - 1 && kreu == 0 || kreu == fundi + 1)
    {cout << "\nQueue eshte i mbushur.";return; }
    else if (fundi == -1)
    {fundi++; kreu++;}
    else if (fundi == size - 1)
        fundi = 0;
    else
        fundi++;

    cout << "Jep vleren per te shtuar ne queue: ";
    cin >> radha[fundi];
}

void queue::largo()
{
    if (kreu == -1)
        cout << "\nQueue eshte i zbrazet"; return;
    cout << radha[kreu] << " u fshi nga queue" << endl;
    if (kreu == fundi)
        {kreu = -1; fundi = -1;}
    else if (kreu == size - 1)
        kreu = 0;
    else
        kreu++;
}
void queue::shfaq()
{
    int t = kreu;
    cout << "Vlerat aktuale ne radhe:\n";
    if (kreu == -1)
    {cout << "\nQueue eshte i zbrazet"; return; }
    while (t <= fundi)
    {
        cout << radha[t] << " ";t++;
    }
    cout << endl;
}

int main()
{
    queue radh; int zgjedhja;
    cout << "Queue (radha):\n";
    do{
        cout << "\n1. Nderfut  - (Inserto)\n2. Fshi  - (Delete)"
            << "\n3. Shfaq - (Display)\n4. Mbyll - (Exit)\n";
        cout << "\nZgjidhni veprimin: ";
        cin >> zgjedhja;
        switch (zgjedhja)
        {
        case 1: radh.nderfut(); break;
        case 2: radh.largo(); break;
        case 3: radh.shfaq(); break;
        }
    } while (zgjedhja != 4);
    return 0;
```

Shembull 7.5

- Të shkruhet programi në të cilin implementohet STL stack. Një tekst I shkruar përmes tastiere të ruhet në stek. Më pas të largohen të gjitha karakteret si dhe të shtypen karakteret e larguara.

```
1  #include <iostream>
2  #include <stack>
3  using namespace std;
4
5  int main()
6  {
7      stack< char> steku;
8      cout << "Sheno nje rresht te tekstit:\n";
9      char ch;
10     cin.get(ch);
11
12     while (ch != '\n')
13     {
14         steku.push(ch);
15         cin.get(ch);
16     }
17     cout << "\nShtypja nga fundi:\n";
18     while (!steku.empty())
19     {
20         cout << steku.top();
21         steku.pop();
22     }
23     cout << endl;
24     system("pause");
25     return 0;
26 }
```

Shembull 7.6

- Të shkruhet programi në të cilin implementohet STL queue dhe përdoren funksionet themelore si **push**, **pop**, **empty**, **front** dhe **back**.

```
1  #include <iostream>
2  #include <queue>
3  using namespace std;
4
5  int main()
6  {
7      queue<int> radha;
8      int x;
9
10     cout << "Jep disa numra te plote (jep 0 per fund):\n";
11
12     do {
13         cin >> x;
14         radha.push(x);   //shtyji vlerat
15     } while (x);
16
17     cout << "radha permban: ";
18     while (!radha.empty()) //deri sa te zbrazet
19     {
20         cout << ' ' << radha.front();//elementi i pare
21         radha.pop();            //terhiq
22     }
23     cout << '\n';
24
25     radha.push(77);         //shtyje
26     radha.push(16);         //shtyje
27     radha.front() -= radha.back();    // 77-16=61
28
29     cout << "radha.front() tani eshte "
30         << radha.front() << '\n';
31     system("Pause");
32     return 0;
33 }
```