

Razvoj softvera

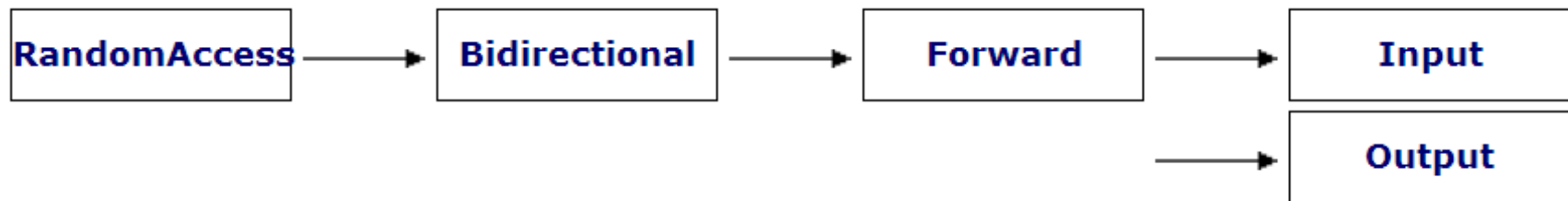
Vežbe 2

Iteratori

- vrste iteratora:
 - običan - omogućava čitanje i menjanje elemenata kolekcije
 - konstantan - omogućava samo čitanje elemenata kolekcije (`const_iterator`)
 - obrnut – omogućava obilazak elemenata od poslednjeg prema prvom elementu (`reverse_iterator`)
 - konstantan obrnut (`const_reverse_iterator`)
 - `rbegin` i `rend`

Iteratori

- Kategorije



Iteratori

category				characteristic	valid expressions
all categories				Can be copied and copy-constructed	X b (a) ; b = a ;
				Can be incremented	++a a++ *a++
Random Access	Bidirectional	Forward	Input	Accepts equality/inequality comparisons	a == b a != b
				Can be dereferenced as an <i>rvalue</i>	*a a->m
		Output		Can be dereferenced to be the left side of an assignment operation	*a = t *a++ = t
				Can be default-constructed	X a ; X ()
				Can be decremented	--a a-- *a--
				Supports arithmetic operators + and -	a + n n + a a - n a - b
				Supports inequality comparisons (<, >, <= and >=) between iterators	a < b a > b a <= b a >= b
				Supports compound assignment operations += and -=	a += n a -= n
				Supports offset dereference operator ([])	a [n]

Algoritam za sortiranje

- `template <class
RandomAccessIterator> void sort
(RandomAccessIterator first,
RandomAccessIterator last)`
 - sortira elemente u rastućem poretku
 - `#include <algorithm>`

Iteratori tokova

- Tokovi - niz elemenata
- `ostream_iterator` – referiše na izlazni tok
 - `ostream_iterator<T>(cout)` – iterator na standardni izlaz
- `istream_iterator` – referiše na ulazni tok
 - `istream_iterator<T>(cin)` – iterator na standardni ulaz
 - `istream_iterator<T>()` – iterator na kraj ulaznog toka
- Ne koriste se direktno, već se prosleđuju kao argumenti funkcijama

Algoritam za kopiranje

- ```
template <class InputIterator,
class OutputIterator>
OutputIterator copy
(InputIterator first,
InputIterator last,
OutputIterator result);
```

  - Kopira elemente u opsegu [first, last) u opseg koji počinje sa result
  - Vraća iterator koji pokazuje na element koji je iza poslednjeg kopiranog elementa
  - ```
#include <algorithm>
```

Funkcija za dodavanje elemenata

- `template <class Container>`
`back_insert_iterator<Container>`
`back_inserter (Container& x);`
 - Funkcija pravi iterator
`back_insert_iterator` za kontejner,
koji omogućava algoritmima da dodaju nove
elemente na kraj kontejnera
 - `#include <iterator>`

Klasa

- Enkapsulacija – prikrivanje interne strukture klase. Korisniku se predstavljaju samo metodi i podaci klase čija je upotreba neophodna za njeno funkcionisanje, dok se ostali članovi sakrivaju.
- vidljivost članova klase
 - `private` – članovi su na raspolaganju samo unutar klase
 - `public` – članovi su raspoloživi svim korisnicima klase
 - `protected` - članovi su raspoloživi samo unutar definicija metoda klase i njenih naslednika

```
protected:  
    int x;  
    string y;
```

Konstruktori

- konstruktori – metodi za inicijalizaciju novih objekata

`Razlomak (...)`

- podrazumevani konstruktor – bez argumenata, sa praznim telom metoda
 - Inicijalizuje sve članove podataka primenom odgovarajućih konstruktora bez argumenata

- podrazumevane vrednosti argumenata – mogu se navoditi u svim funkcijama i metodama klase, osim u operatorima

```
Razlomak( int b=0, int i=1 )
```

- lista inicijalizacija

```
Razlomak( int b=0, int i=1 )  
: _Brojilac(b),  
  _Imenilac(i)  
{ }
```

Operatori

- aritmetički operatori

```
Razlomak operator + ( const Razlomak& y )  
const
```

- operatori poredjenja

```
bool operator==( const Razlomak& x )  
const
```

- operatori inkrementiranja

```
Razlomak& operator++ ( )  
Razlomak operator++ (int)
```

- operator konverzije

```
operator double( ) const
```

Operatori za čitanje i pisanje

```
ostream& operator << (ostream&  
str, const Razlomak& r)
```

```
istream& operator >> (istream&  
str, Razlomak& r)
```

Prijateljske klase i funkcije

- prijateljske klase i funkcije neke klase imaju pristup svim privatnim i zaštićenim elementima te klase

```
friend class <ime klase>;
```

```
friend class <deklaracija  
funkcije>;
```