

# Razvoj softvera

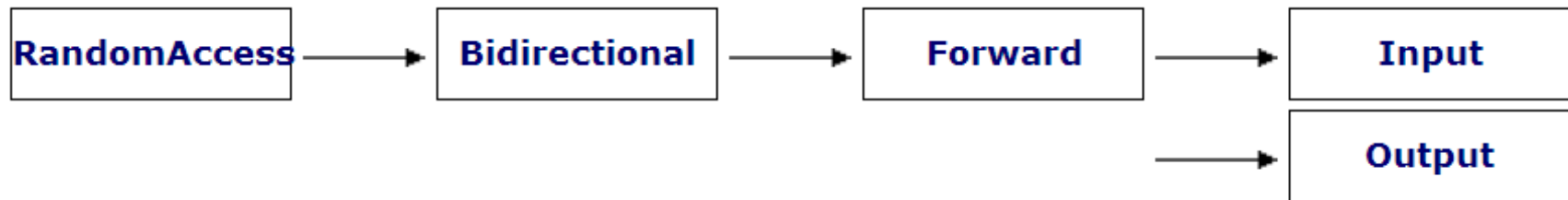
## Vežbe 1

# 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>`