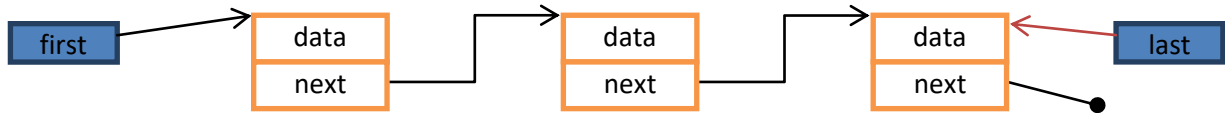


Algorithms and computational complexity

Preparation for the laboratory 4: Linked lists.

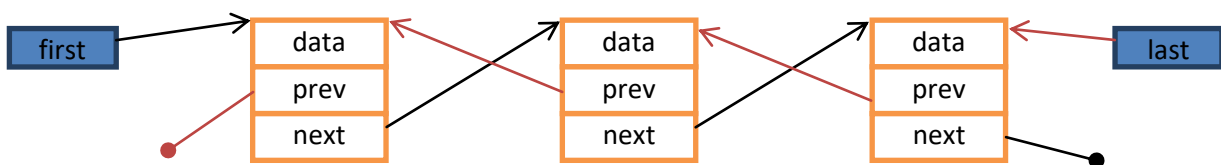
1. One-way linked list concept.

A one-way linked list is a data structure that allows you to store data in an orderly form, as well as to quickly insert and delete items to and from the list. The list consists of elements containing data and a pointer (or reference) to the next element. This way, the class representing a list stores only the reference to the first element of the list (optionally also the last one) in order to remember the entire list. The structure of the one-way list is illustrated in the figure below:



2. Two-way linked lists.

Two-way lists are ordered, linear data structures. They differ from one-way lists only in that the navigation between elements of the list can be performed in two directions. A typical bi-directional list consists of elements that, in addition to the data stored in the list, contain information about the previous and next elements of the list. The structure of the two-way list is illustrated in the figure below:



3. Generic classes.

Generic classes are classes with parameterized data types. Generic classes have a complete implementation, but they do not define the data types used in this implementation. For example:

```
public class Element<T> {
    private T data;
    private Element<T> next, prev;
    . . .
}
```

The parameter of the Element class is some type T. We can now use this class instance to store different types of data. E.g. (after defining the appropriate constructor):

```
Element<Integer> e = new Element<Integer>(5);
Element<String> e2 = new Element<String>("Ala ma kota");
```

Since Java 7, the Java compiler has a built-in type inference mechanism for deduction of generic types. Thus, we do not have to repeat parameters (types) each time. Therefore, the following expression is valid:

```
Element<Integer> n = new Element<>(5);
```

The generic programming mechanisms available in Java should be used.

Attention! The types given as parameters must be reference types! E.g. use *Integer*, not *int*.

4. Tasks.

1. Implement a one-way linked list. Use Element-like objects to store data with a reference to the next element, similar to a stack. The list should allow adding elements to the beginning and end, deleting the first and last element, and writing the list contents onto console.
2. Implement a bidirectional list that allows the same operations as in the Task 1.
3. In the class representing the list, implement the iterator () method returning the object implementing the Iterator <T>.

<https://docs.oracle.com/javase/7/docs/api/java/util/Iterator.html>