ID1020 2018-09-04 #2

Algorithm: Set of well-defined instructions to solve a problem
Correct if the procedure stops and returns the desired value for all inputs.

Stack: first in last out queue: first in first out
Insert, Remove, Iterate

Implementation → Interface → Client
Transparent means the client is seperated from the implementation from the interface.

- The client does not need to know the implementation
  ex. typedef int pris;
- The implementation does not need to know the clients requirements.

Example API:
Stack: create, push, pop, isEmpty, size

Linked list: Node — value and pointer to next element (reference) (node)

Implementation of Stack using Linked list (see slides)
Access time: constant worst case
Memory overhead: ~40 bytes per stack node
Implementation using array (see slides)
Problem: stack overflow when array is filled

Loitering (söla) occurs when we keep an unused reference
Garbage collector cannot reclaim memory

Validation - testing
Verification - proving

Resizing array at every push/pop is too expensive (N^2)
Instead we can double the size whenever a bigger array is needed (N)
Halve array size when it is a quarter full.
Best, Worst and Amortized time

Different implementations using same interface can be used by the same client code.

Queue: create, enqueue (insättning), dequeue (uttag), isEmpty, size

Implementation using array, index modulo array size

Generics "Something that works for different types"
type parameter instead of specific data type.
Generic arrays do not exist in Java because of backwards compability
primitive data types cannot be used with generic. We need to use wrapper objects (autoboxing)
Wierd behaviour when using objects as primitive data types.

Iterators used to iterate through objects

implement java.lang.Iterable

hasNext, next, (remove) ← use at own risk

Bags Applications (tomorrow)