Data Structures 2018
Exercise 4 (Week 40)

- **Notice that based on university's new regulations on degrees, you can have a degree fail from the course which is put to the register.**

  If a student does not participate in the course and does not cancel his/her enrollment, or if he/she discontinues the course, he/she will be assigned a fail grade for the course in question.

- **Students who participate in exercise group must be in place before the exercise group begins (12.15/14.15/16.15). Students who come late do not get the exercise points.**
- **Check the numbers of exercises made before you come to the exercise group. By this means we can save a lot of time when filling the exercise point list.**
- **Notice that pseudocode does not mean the same this as Java code. Pseudocode is not a programming language dependent presentation for an algorithm.**

1.-2. In this task you need *DynArrStack.java* (stack implementation using a dynamic array) and *DynArrStackTest.java* (a test program) files. Implement the following two operations into the file DynArrStack.java:

`push(object x)`: adds object x on top of the stack. If the array already contains $n = N$ items, the stack is first moved into a new, twice as large array (meaning that $N \leftarrow 2N$).

`pop()`: removes and returns the top item of the stack. If the array contains $n = \lfloor N/4 \rfloor$ items *after* the removal and $N \geq 2$, the stack is moved into a new array that has half size (ie. $N \leftarrow N/2$).

The notation $\lfloor y \rfloor$ means rounding the value y towards zero (so-called floor-function). Java automatically rounds integers in this manner, so you do not need to do explicit rounding. Test your implementation by running the program DynArrStackTest.java (note: the test program does not function correctly unless you first implement the above operations).

3. A reverse polish notation (RPN) calculator uses a stack. The calculator works as follows: Entering a number puts it on top of the stack and entering an operation op causes the topmost number x and second topmost number y to be popped off the stack and the number y op x to be pushed on the stack. Thus an operation *xopy* is entered as *xyop*. For example $1 + 2$ is computed with "1 2 +" and $(2 * 3) - 4$ is computed with "2 3 $*$ 4 $-$". How do you calculate the following expressions with a RPN calculator ?
   a) $1 + 3 + 5 - 7$ c) $3 * (2 + 4 * 3)$
   b) $(6 - 3) * 2 + 1$ d) $(3 + 4) * (20 - (3 * 4 + 2))$
4. Describe in pseudocode an algorithm, which returns the second to last node of a singly-linked list, when the member variable next of the last node is null.
5. Describe in pseudocode the algorithm `merge`, which gets as input two singly linked lists A and B sorted in ascending order and returns a sorted list, which contains the elements of both lists in ascending order. E.g. if $A = (2, 5, 6), B = (1, 3)$ then the result is $(1, 2, 3, 5, 6)$.
6. Describe in pseudocode the algorithm ReversePart, that gets as input a singly linked list L and indices x and y $(0 \leq x < y)$. The algorithm reverses the order of the elements between the given indices, (including the elements at indices x and y) and returns the resulting list. If $L = (a, b, c, d, e, f), x = 1, y = 4$, the result is (a, e, d, c, b, f). (The index of the first element of the list is 0).
7. Describe in pseudocode an algorithm `IsGreater(`A`,`B`)` that gets as input two integers A and B expressed in linked lists and returns a boolean value `true/false` that tells whether $A > B$. The integers are expressed so that the first element of the list is the least significant digit of the integer. E.g. if $A = 986$ and $B = 674$, the corresponding linked list forms are $A = (6, 8, 9)$ and $B = (4, 7, 6)$, and the algorithm returns the value `true` as in this case $A > B$. A negative number is expressed by adding the item '-' to the end of the list. E.g. the number -175 is expressed in the form $(5, 7, 1, -)$.

8. For the following tree, what are the

(a) i. root node?

ii. leaf nodes?

iii. internal nodes?

iv. What is the parent node of C?

(b) For the node B, what are the

i. children?

ii. ancestors?

iii. descendants?

iv. siblings?