Lecture 11 - Recursion

Housekeeping

- Plan for the rest of the course
  - Today: recursion
  - Next week: generics and collections
  - Next week+1: practical programming
    - Send me your requests!
  - Final lecture: course recap
    - Send me your requests!

Housekeeping

- Easter re-exams
  - Part A only
  - 8/4, SB building
  - **NOT** the huge SB multisal!
  - Register in Ladok, starting from 11/2
  - For extended time, etc: contact tentamen.stodet@chalmers.se
  - You can **NOT** lose points at the exam review!

Quiz time!

qui.su/4NfF

OOP recap

- If class A extends class B, then:
  - Every public method and instance variable of B, is now also part of A
  - We can use an object of class A as though it were of class B
  - Constructors in A may use `super(…)` to chain constructors in B

OOP recap

- If class A extends class B, then:
  - A can *override* methods from B
  - That is, if a method `m` is defined in B, A can provide its own version of that method.
  - A class which overrides a method `m` can use `super.m(…)` to call its superclass' version of `m` instead of its own

OOP quiz time!

qui.su/4NfF

Example - Factorial

The function n! (the **factorial** of n) is defined to be:

n! = 1 . 2 . … . n

We can define this recursively:

0! = 1

(n+1)! = n! * (n+1)

How Does It Work?

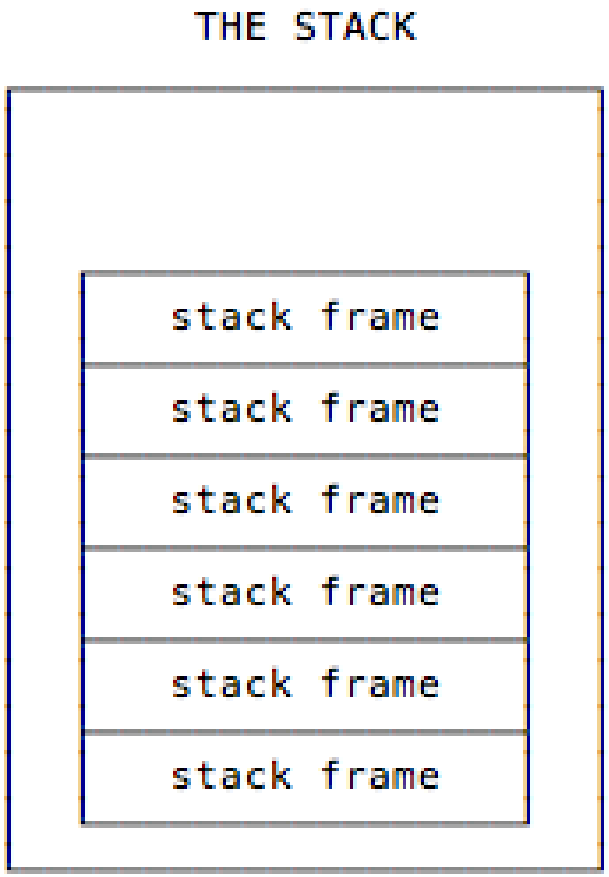


Image from
https://alvinalexander.com/scala/fp-book/recursion-jvm-stacks-stack-frames

Example – number of zeros in (the decimal representation of) a number

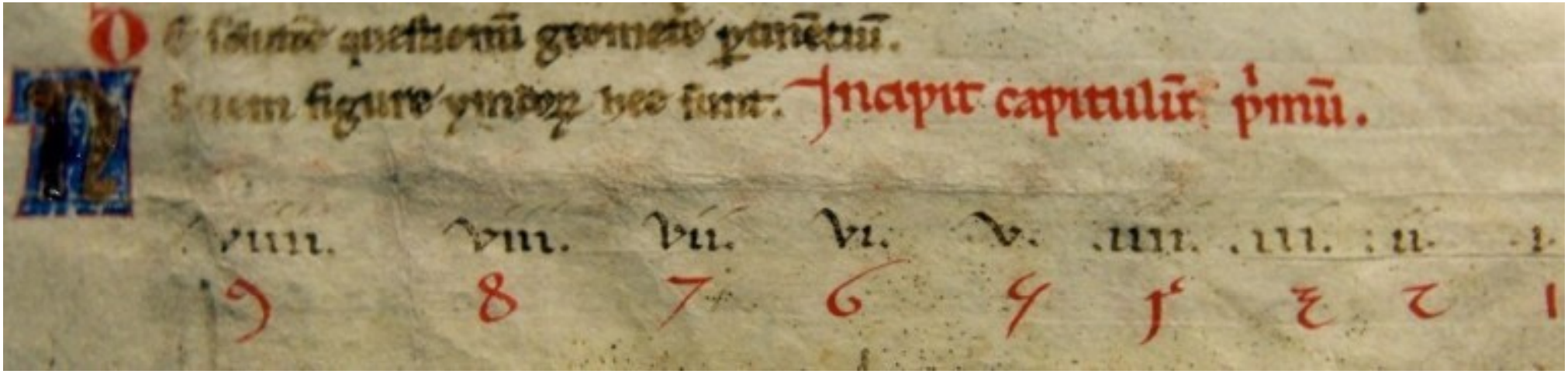
Iteration vs Recursion – Exponentiation

Recursion vs Iteration

- It is always possible to rewrite a recursive function so that it is not recursive.
- Iterative methods are usually faster and use less memory
- Recursive methods can be easier to read, modify, test and debug
- Very useful for “backtracking” solutions

Fibonacci Numbers

Leonardo of Pisa (“Fibonacci”) wrote *Liber Abaci* in 1202:



Fibonacci Numbers

One of the puzzles from *Liber Abaci*:

A newly born pair of rabbits, one male, one female, are put in a field;

rabbits are able to mate at the age of one month so that at the end of its second month a female can produce another pair of rabbits;

rabbits never die and a mating pair always produces one new pair (one male, one female) every month from the second month on.

How many pairs will there be in one year?

Fibonacci Numbers

Month	Young pairs	Adult pairs
1	0	1
2	1	1
3	1	2
4	2	3
5	3	5
6	5	8
7	8	13
8	13	21
9	21	34
10	34	55
11	55	89
12	89	144

Fibonacci Numbers

$$F_1 = 1$$

$$F_2 = 1$$

$$F_{n+1} = F_{n-1} + F_n$$

Break + quiz time!

qui.su/4NfF

Towers of Hanoi

https://www.mathsisfun.com/games/towerofhanoi.html

Divide-and-Conquer

To solve a problem on a big case:

Just assume you know how to solve it on a smaller case!

Example: Binary Search

To find an element n in a **sorted** array a[]:

- If a[] is empty, fail
- Compare n to the middle element a[i]
- If they are equal, return i
- If n is smaller, **find n in the subarray a[0],…,a[i-1]**
- If n is larger, **find n in the subarray a[i+1], …, a[a.length-1]**

Towers of Hanoi
Can we implement an algorithm for it?
https://www.mathsisfun.com/games/towerofhanoi.html

How to Write a Good Recursive Method

- The method should contain an if-statement
- One branch of the if-statement (the base case) returns without recursing
  - There can be more than one
- The other branches involve recursive calls in which the parameter is smaller in some sense.

The Compiler/IDE Cannot Help You!

- It is impossible to write a program that will decide correctly whether any recursive function always terminates
  - This is an instance of the Halting Problem

Reading and Exercises

- **Reading**
  - *Java Direkt med Swing* 19.4
- **Exercises**
  - *Java Direkt med Swing* exercises 19.5, 19.6
  - Bonus Exercises: Sorting, fix the bug(s) in this lecture’s code