**Programmering 2 – Noter**

**Kontrolstrukturer**

*Sekvens* – Klasser, objekter metoder. Afgøre hvilken rækkefølge der køres.

*Selektion* – If/else

*Løkker* – For/while/For each

**Nedarvning**

Superklasser og subklasser

Generalisering og specialisering

**Typer i java**

Simple – int, char, double (lille begyndelsesbogstav) Pass by value

Abstrakte – String mm. (Stort Begyndelsesbogstav) Pass by reference

* Klasser (Konkrete)
* Abstrakte klasser
* Interface
* Enum

**Comparable**

Comparable er en metode til at se om noget er større, mindre eller lige med. Resultere i negativ, positiv eller 0 integer.

Bruges på et objekt eller variable med samme slags som input.

Skal implementes på klassen. Derved tvinges man til at lave comparaTo() metoden.

**Lambda funktion**

En funktion er en metode der ikke er bundet til en klasse. Hvilket metoder normalt er.

Lambda funktion: ’->’

**Abstract og interface**

Abstract klasse kan ikke oprettes. Attributter og metoder arves til ’børne’-klasser

Interface – Klasse kan have mange interfaces koblet på sig. For normalt kan en klasse kun arve fra én anden klasse.

**Test**

Test metoder må gerne have redundant kode. I modsætning til generel konvention.

Triple A notation

-Arrange – Opstil testdata  
-Act – Kald metoden der skal testes  
-Assert – Test påstande på returværdi

**Collections**

Polymorfi er ligesom flexibilitet

**Kædestrukturer**

Vigtige elementer om datastrukturer:  
Handler om at samle objekter, så de kan gemmes og fremfindes effektivt.

Datastruktur kerneelementer:

\*Intern struktur – Enten array eller kædestruktur

\*Input/output (I/O) - Hvordan tilføjer/fjerner objekter

\*Performance – Big O notationen - Handler om hvad koster det at gøre noget. Fx add, remove, count.

Big O table:

Et billede, der indeholder tekst, skærmbillede, nummer/tal, Font/skrifttype

Automatisk genereret beskrivelse

Stak

**Generics**

Gør koden mere typestrærk

Fx public class Cup<T> {}

T gør Klassen generisk

private String getName ( . . . )

Visibility, returntype, name, parameter  
Derudover kan tilføjes modifiers som: static, <E>, final

**Rekursion**

Det er hvis en metode kalder sig selv.

Består af: termineringsregel(noget der stopper reskusionen) og rekurrensregel(her kalder metoden sig selv)

Iteration vs rekursion – Fordle kan være at rekursion kan være ismplere i nogle tilfælde