

Eksamensnoter af Ahmad Alkaseb

IDE = Integrated Development Environment

Begreb	Forklaring	Eksempler i Java
Syntax	<i>Syntax er programmets (IDE) regler.</i>	; Case sensitive
Keyword	<i>Alle programmer (IDE) har deres egne keywords.</i>	class new
Kommentarer	<i>Alle programmer (IDE) har hver deres måde at skrive kommentarer på.</i>	<i>/* */</i> <i>//</i>
Erklæring af variabler	<i>Det er nødvendigt at bruge variabler til at gemme værdier, men nogle gange har man ikke værdien endnu, og derfor kan man nøjes med at erklære variabelen.</i>	String s int i float f
Initialiser	<i>At initialisere en variabel betyder at give den en startværdi.</i>	s = " " i = 0 f = 0,01
Tildele	<i>At tildele en variabel en værdi er noget man gør efter man har initialiseret den.</i>	s = "Hej" i = 10 f = 10,01
Scopes	<i>Der findes to forskellige slags scopes. Globale- og lokale scopes.</i>	{ Globale {Lokale} }
Datatyper	<i>En datatype er en variables type. Det vil sige er variabelen en tekst, et heltal, et kommatal osv.</i>	String int float
Udtryk	<i>Udtryk er begreb man bruger, når man taler om koden.</i> <i>Det kan være for eksempel:</i>	Tildelings udtryk Boolean udtryk (Brugt i parenteserne for statements)

Concatenation	<i>Et begreb man bruger, når man sætter to String sammen, og evt gemmer den i en ny variabel.</i>	String 1 = "hej " String 2 = " med dig" String 3 = 1 + 2
Compile time	<i>Dette er noget, der sker løbende, mens man skriver ens kode. Det kan være, at der mangler parenteser, semikoloner osv.</i>	
Compile run time	<i>Dette er noget, der sker efter man kører ens program.</i>	Det kan for eksempel være, når man prøver at få fat i en index plads i en array, som ikke findes.
Statement	<i>En linje kode.</i>	
Logiske operatorer	<i>Disse logiske operatorer bruges i boolske udtryk.</i>	! &&
Algoritme	<i>Dette er alt ens kode, medmindre man specificerer det til en mindre del. Det er altså koden, men ikke programmet!</i>	
Betingelser	<i>En betingelse er et boolsk udtryk som man bruger i kontrolstrukturer</i>	if (5 > 0)
Evaluering	<i>At vurdere om betingelsen er opfyldt eller ej, hedder at evaluer.</i>	Vores ovenstående eksempel er derfor opfyldt, da 5 er større end 0.
Iteration	<i>At iterer betyder at gentage koden indtil en betingelse er opfyldt.</i>	Dette sker for eksempel i for- eller while loops.
Increment / decrement	<i>Dette betyder at enten tilføje en eller trække en fra.</i>	int i = 0 i++ i--
Aritmetisk operator	<i>Dette er vores matematiske tegn, som kendes fra matematikken.</i>	+ - * /

Metoder	<i>Metoder skrives for ikke at gentage kode.</i>	
Metodekald	<i>Metodekald sker, når vi kalder på metoden.</i>	
Returtype	<i>En metode kan enten returnere en værdi eller ikke.</i>	String int void
Parametre	<i>En metode kan også i nogle tilfælde have brug for nogle argumenter. Derfor skal vi fortælle i metode signaturen, at den har nogle parametre.</i>	
Argumenter	<i>Argumenter er altså værdier, som vi smider ind i en funktion.</i>	
Metodesignatur	<i>Hvad hedder den. Hvilken returtype. Hvilke parametre tager den. Det er altså hele første linje.</i>	
Klasser	<i>Vi laver så mange klasser, når vi skriver i Java. Dette gør vi for at få et større overblik. Dette kaldes også objektorienteret programmering.</i>	
Instantitere	<i>Dette betyder at oprette en instans af en klasse.</i>	
Magiske numre	<i>Magiske numre er værdier, som vi bruger uden at gemme i en variabel.</i>	
At gemme information	<i>Dette sker i klasser, når vi bruger private instansvariabler. Således gemmer vi den information, som andre ikke kan se.</i>	
Getter eller Setter metoder	<i>En metode, som bruges til at sætte værdien for en instansvariabel og en til at få fat i værdien, eftersom den er sat til privat.</i>	

Access modifiers	<i>Modifiers bruger vi i klasserne til at gøre nogle af instans variablerne eller metoderne tilgængelige eller utilgængelige</i>	public privat protected
Overloading	<i>Når man opretter flere metoder af samme navn, men med forskellige parametre.</i>	
Try-Catch-Finally	<i>Dette er noget man prøver, og hvis det lykkes, kaster den en fejl tilbage, som bliver fanget.</i>	
Datastrukturer	<i>Der findes mange, men et eksempel kunne være en array.</i>	
Traversal	<i>At loope igennem en datastruktur, som i for eksempel en array.</i>	
Alias	<i>En variabel som refererer til den samme værdi eller objekt, som en anden.</i>	
Flag variabel	<i>Et flag variabel er et boolsk udtryk, som hjælper os med at holde styr på kode flowet eller udføre bestemte handlinger baseret på bestemte betingelser.</i>	Forestil dig, at vi skal holde øje, om en person er logget ind eller ej. Dette kan vi gøre med en flagvariabel, som hedder isLoggedIn (som er sat til false indtil person logger ind)
Entity	<i>Entity-klasser i Java refererer normalt til klasser, der repræsenterer objekter eller enheder i den virkelige verden.</i>	public class Human public class Car public class Road
Utility	<i>Utility-klasser i Java refererer normalt til klasser, der indeholder en samling af hjælpefunktioner eller metoder.</i>	public class InputOutput (Hjælper med at skrive og få input fra brugeren)
Client	<i>Client-klassen er typisk Main-klasssen. Denne bruges til at køre programmet.</i>	

Static	<p><i>En static metode er en metode, som kan bruges uden at oprette en instans af en klasse.</i></p>	
List	<ul style="list-style-type: none"> • <i>Har en rækkefølge</i> • <i>Har index tal</i> • <i>Har dubletter</i> <p><i>Der findes to forskellige lister af List.</i></p> <p><i>ArrayList, LinkedList og flere.</i></p> <p><i>Man vælger altid den rigtige liste ud fra dens performance tid.</i></p> <p><i>De har forskellige yde egenskaber (performance). Dette kan slås op, når man skal bruge den.</i></p>	
Set	<ul style="list-style-type: none"> • <i>Ingen rækkefølge</i> • <i>Ingen index numre</i> • <i>Ingen dubletter</i> <p><i>Definitionen af dubletter kan defineres.</i></p> <p><i>HashSet, TreeSet og flere.</i></p> <p><i>De har forskellige yde egenskaber (performance). Dette kan slås op, når man skal bruge den.</i></p>	
Map	<ul style="list-style-type: none"> • <i>Par af key & value</i> <p><i>Key er unik.</i></p> <p><i>Value må godt være ens.</i></p> <p><i>HashMap, HashTree og flere.</i></p> <p><i>De har forskellige yde egenskaber (performance). Dette kan slås op, når man skal bruge den.</i></p>	

LIFO	<i>Last in first out</i>	
FIFO	<i>First in first out</i>	
DRY	<i>Don't repeat yourself</i>	
SoC	<i>Separation of concern.</i> <i>Dette begreb handler om at afgrænse ansvarsområder for metoder og klasser.</i>	