

Praktisk information

Den obligatoriske opgave er nu på BlackBoard -> Course Materials -> OOP søjlen -> Obligatoriske Opgaver -> [DynWebJava Obl Opg 1 E17 \(1\).pdf](#).

Emner til lektion 2 (13/9)

Afslutning på terning eksemplet.

Mere om klasser og objekter.

Start på den obligatorisk opgave.

Forberedelse til lektionen

Løsningen til terning opgaven, som vi ikke blev færdige med under lektion 1, ligger på BB -> [OOP Lektion 01](#) -> [dice.zip](#).

Pak de 3 klasser ud og læg dem i et NetBeans projekt under src\dice.

- Die.java: Den generelle terning, vi arbejde med
- DiceCup.java: Et rafflebæger med 2 terninger
- DiceStatistic.java: Opsamler hvormange gange der slås 2, 3, 4, osv med rafflebægeret, i et array. Bemærk at 2'erne tælles op i index 0, 3'erne i index 1 osv, da det laveste antal øjne er 2, når man slår med 2 terninger. Vedrørende input fra tastaturet: Se slides 50-56 i ppt til kapitel 2, samt javaDoc for java.util.Scanner.

Studér koden, og prøv at forstå hvad der foregår. Eksperimenter lidt med at lave små ændringer.

Bearbejd følgende fra kapitel 3 i Java Foundations:

- *Afsnit 3.1* skal læses grundigt. Det er vigtigt at forstå forskellen på om en variabel indeholder en af Javas *primitive* typer eller en *reference* (pointer) til et objekt.
- *Afsnit 3.2* er en kort introduktion til klassen *String*. Den skal vi arbejde en del med i hele kurset, så skim afsnittet igennem. (En mere fyldestgørende beskrivelse kan ses i [Liang ch9 Strings.pdf](#). Den kan løbende benyttes til opslag).
- *Afsnit 3.3* skal læses grundigt
- *Afsnit 3.4* og *3.5* benyttede vi lidt fra under terning-øvelsen.
- *Afsnit 3.6* og *3.7* tror jeg ikke umiddelbart relevante for jeres uddannelse; men skim dem lige igennem.
- *Afsnit 3.8* er vigtigt.

Løs *Programming Projects* begyndende på side 106: *PP3.1* og *PP3.2*

Bearbejdelse af dagens emner

Hurtig gennemgang af JF chap. 2.

Hvis der er uklarheder til stoffet i kap 2 (PowerPoint i [java foundations slides.zip](#)), tagerv i dem nu:

- Strings, concatenation, escape
- Deklaration og brug af variable
- Primitive typer
- Expressions, arithmetiske operatorer, casting
- *Scanner*-klassen (den nåede vi ikke at snakke om sidst, så vi kigger under alle omstændigheder på den)

Et par små opgaver i klassen:

PP 2.2 Write an application that reads three integers and prints their average.

PP 2.3 Write an application that reads two floating-point numbers and prints their sum, difference, and product.

PP 2.10 Write an application that prompts for, reads an integer representing the length of a square's side, and then prints the square's perimeter and area.

Så ser vi på kap 3:

- PowerPoint til afsnit 3.1. Spørgsmål afklares undervejs.
- String-klassen. Specielt ser vi på metoderne på fig. 3.1 side 81. Eksempel 3.1 side 82 indlæses i NetBeans og eksperimenterer lidt med det.
- JavaDoc og begrebet *package*.
- Package *java.lang* og *java.util* undersøger vi nærmere.
- De studerendes løsninger til PP3.1 og PP3.2 diskuteres.
- I klassen: PP3.5 (*overflade og rumfang af en kugle*).

Vi følger op på raflebægeret fra sidst og begynder på [den obligatoriske opgave](#).

Emner til lektion 3 (20/9)

Loops og Conditions.

Definition af egne klasser.

Forberedelse til næste lektion

Diverse *if-else*, *switch*, *while* og *for*-strukturer burde i hovedtræk være kendt fra fx JavaScript. Kig PowerPoint præsentationen til [kap 4](#) igennem og notér ned, hvis der er emner der skal uddybes.

NB: Slide 34 – 43, *Comparing Data* er det nødvendigt at i kender.

Slide 68 – 72, *Iterators* har i sandsynligvis ikke før stødt på. Det er ikke vigtigt lige nu, hvorfor slide 88 – 89 heller ikke er vigtige.

Studer kap 5, afsnit 5.1 – 5.5 meget grundigt. Indlæs eksemplerne fra teksten i NetBeans og eksperimenter med dem.

Løs PP5.1 og PP5.2 side 238 (den Coin-klasser skal benyttes er *Listning 5.4* side 186), samt PP5.4.

Afsnit 5.6 skal blot skimmes. Dog skal underafsnittet om *this*-referancen side 212-213 kendes.

Afsnit 5.7 og 5.8 skal blot skimmes.

Kom så langt som muligt med 1. del af den obligatoriske opgave.