

Programming Basics Medley

<https://processing.org/tutorials>

1 Generelt

. Hvad er en compiler?

En compiler er den der laver sin kode om til machine code som computeren bruger til at udføre processen.

2 Pseudokode

. Hvad er pseudokode?

Pseudokode er hvor man laver sit stykke kode i en meget sympel form for at få et overblik over hvad det skal.

. Hvorfor kan det vare smart at anvende pseudokode?

Man for et overblik over hvad koden skal kunne og det gør den nemmer at forklare i simple form til andre.

3 Kommentar

. Hvilke muligheder har du for at indsatte kommentarer i din programkode?

Der er mange forskellige måder at indsætte kommentatere angående af hvad sprog man bruger i java er det `//` eller `/*` mens if x. Python er det `#`

4 Rutediagram (FlowChart)

. Hvad er et rutediagram, og hvad bruges det til?

Et flowchart bruges til at lavede system om hvad encode skal gøre trin for trin

. Hvilke elementer kan et rutediagram bestå af? (de meste basale elementer)

Start, slut, if else, blok

5 Fejlfinding (Debugging)

. Hvordan vil du fejlfinde i et program som ikke opfører sig efter hensigten?

Brug breakpoints eller `println()` til at finde ud af hvad der går galt, hvis ikke det virker kan jeg google fejl koden i console

6 Console IO

Skriv til, og lasning fra konsol.

- . Giv bruger en besked om at indtaste sit navn.
- . Gem navnet i en variabel af passende type.
- . Giv bruger en besked om at indtaste et heltal.
- . Gem heltal 1 i en variabel af passende type
- . Giv bruger en besked om at indtaste endnu et heltal
- . Gem heltal 2 i en variabel af passende type
- . Beregn produktet af de to indtastede heltal
- . Giv brugeren besked (med navn) om hvad produktet af de to heltal blev.
- . Udvid program således beregning af produktet foretages i en funktion som du selv opretter.

```
import controlP5.*;
ControlP5 text;

void setup() {
    size(340, 500);
    background(51);
    textAlign(CENTER);
    PFont font = createFont("Arial", 20);
    text = new ControlP5(this);

    text.addTextfield("Username").setPosition(20, 100).setSize(200,
40).setFocus(true).setFont(font).setAutoClear(false);
    text.addTextfield("Heltal1").setPosition(20, 200).setSize(200,
40).setFont(font).setAutoClear(false);
    text.addTextfield("Heltal2").setPosition(20, 300).setSize(200,
40).setFont(font).setAutoClear(false);
    text.addBang("Submit").setPosition(240, 300).setSize(80,
40).getCaptionLabel().align(ControlP5.CENTER,
ControlP5.CENTER).setFont(font);
}
void draw(){}

void Submit() {
    String username = text.get(Textfield.class,"Username").getText();
    float heltal1 = float(text.get(Textfield.class,"Heltal1").getText());
    float heltal2 = float(text.get(Textfield.class,"Heltal2").getText());
```

```
if (heltal1 %1 != 0) {
text.getText(Textfield.class,"Heltal1").setText("invalid");
}
if (heltal2 %1 != 0) {
text.getText(Textfield.class,"Heltal2").setText("invalid");
}

heltal1 = float(text.getText(Textfield.class,"Heltal1").getText());
heltal2 = float(text.getText(Textfield.class,"Heltal2").getText());

int product = int(heltal1) * int(heltal2);
text(username + " your product is " + product, 170, 450);
}
```

7 Lokker

- . Lav en forlokke der skriver tallene fra og med 0 til og med 99 ud til console
 - . Lav en forlokke der skriver tallene fra og med 1 til 10 ud i til console
 - . Lav en forlokke der skriver 7 tabellen ud i consollen
 - . Lav to forlokke. Den første forlokke skal kore 5 gange. I kroppen af denne forlokke laves endnu en forlokke som gentages 10 gange. I den indre forlokke udskrives produktet af de to variabler som du bruger til at styre dine lokker med til consol. Inden du korer programmet, hvad forventer du at se ?
- jeg forventer at se nogle af tabellerne fra 1-5

```
void setup() {  
  
    // prints numbers 0-99  
    for (int i = 0; i < 100; ++i) {  
        println(i);  
    }  
    println("_____");  
    // prints numbers 0-10  
    for (int j = 0; j < 11; ++j) {  
        println(j);  
    }  
    println("_____");  
    // prints 7 tables from 0-70  
    for (int t7 = 0; t7 < 11; ++t7) {  
        println(t7*7);  
    }  
    println("_____");  
  
    // the inner loop prints 10 numbers per increment of the outer loop  
    for (int x = 0; x < 5; ++x) {  
        for (int y = 0; y < 10; ++y) {  
            println(x*y);  
        }  
    }  
    // closes the program when done  
    exit();  
}
```

8 Arrays

. Hvad er et array ?

Et array er en liste med given langde af enten numre eller strings

. Hvordan opretter/erklarer man et array?

Man opretter et array med fx. `Int[]` hvor det er `[]` der siger at der er et array

. Hvordan tilgår man elementerne i et array?

Man tilgår elementerne i sit array med fx. `Array[0]` hvor man får det første element i arrayet

. Opret single array af typen `int` med en størrelse på 10.

. Sat manuelt vardierne 4,8,10,9,3,5,1,6,7,2 ind i dit array.

. Byt om på vardierne på index 3 og index 7.

. Lav en for-lokke som udskriver dit array til consol. (du kan finde størrelse på et array ved: `arrayNavn.Length`)

. Brug forlokke når du beregner summen af alle elementer i array, og udskriv resultat til consol.

. Opret nu et todimensionelt 5x6 heltals array

. Lav nu to indlejrede forlokker som fylder array med random vardier mellem 1 og 65535.

(udskriv array til console)

```
int[] numbers = {4,8,10,9,3,5,1,6,7,2};
    println(numbers);
    println("_____");

int temp = numbers[7];
numbers[7] = numbers[3];
numbers[3] = temp;
println(numbers);
println("_____");

int sum = 0;
for (int i = 0; i < numbers.length; ++i) {
    println(numbers[i]);
    sum += numbers[i];
}
println("summen er " + sum);
println("_____");

int[][] arr2D = new int[5][6];
```

```

for (int i = 0; i < arr2D.length; ++i) {
    for (int j = 0; j < arr2D[i].length; ++j) {
        arr2D[i][j] = round(random(1, 65535));
        print(arr2D[i][j] + " ");
    }
    println();
}
exit();

```

9 Extending the software

I forlængelse af din nuværende kode.

- . Lav et todimensionelt array af typen int, der afspejler en kryds og bolle spilleplade. (eg. 3x3 felter)

- værdien 0 angiver den tomme plads.

- værdien 1 angiver hvid brik

- værdien 2 angiver den sorte brik.

- . lav en funktion som skriver spillepladen ud til consol.

- . lav en funktion som kan flytte en brik fra en position til en anden. (der returneres true ved gyldigt trak, false ved ugyldigt trak.)

- . Konstruer dit program således at du via bruger input fra console kan flytte brik, og efterfølgende vise spilleplade.

```

int[][] grid = new int[3][3];
int player = 1;

void setup() {
    println("Player 1 turn");
    size(300, 300);
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            grid[i][j] = 0;
        }
    }
}

void draw() {

```

```

strokeWeight(2);
for (int i = 0; i < 3; i++) {
  for (int j = 0; j < 3; j++) {
    rect(i*100, j*100, 100, 100);
    if (grid[i][j] == 1) {
      line(i*100+25, j*100+25, i*100+75, j*100+75);
      line(i*100+75, j*100+25, i*100+25, j*100+75);
    } else if (grid[i][j] == 2) {
      ellipse(i*100+50, j*100+50, 50, 50);
    }
  }
}
if(checkWin()){
  println("Player " + (1 + player % 2) + " has won");
  delay(1000);
  for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
      grid[i][j] = 0;
    }
  }
  println("Player " + player + " turn");
}
}

void mousePressed() {
  int x = mouseX / 100;
  int y = mouseY / 100;
  if (grid[x][y] == 0) {
    grid[x][y] = player;
    if (player == 1) {
      player = 2;
      println("Player 2 turn");
    } else {
      player = 1;
      println("Player 1 turn");
    }
  }
}

```

```

    }
}

boolean checkWin() {
    // check rows
    for (int i = 0; i < 3; i++) {
        if (grid[i][0] == grid[i][1] && grid[i][1] == grid[i][2] &&
grid[i][0] != 0) {
            return true;
        }
    }

    // check columns
    for (int j = 0; j < 3; j++) {
        if (grid[0][j] == grid[1][j] && grid[1][j] == grid[2][j] &&
grid[0][j] != 0) {
            return true;
        }
    }

    // check diagonals
    if (grid[0][0] == grid[1][1] && grid[1][1] == grid[2][2] && grid[0][0]
!= 0) {
        return true;
    }
    if (grid[0][2] == grid[1][1] && grid[1][1] == grid[2][0] && grid[0][2]
!= 0) {
        return true;
    }
    return false;
}

```