

Basale programmeringsteknikker: Arrays, Foreach,

▼ Array

Data struktur, hvor alle elementer har en bestemt datatype:



Arrays indekseres altid **startende fra 0**

Positionen kaldes et **index**

Erklæring af array, hvor den gemmer 12 pladser i hukommelsen:

```
months = new int[12];
```

Eksempel på array:

```
public static void main(String[] args) {  
    int index = 5; //den måned jeg vil finde antal dage i.  
    //declare string array og dets værdier  
    String[] months = { "January", "February", "March", "April", "May",  
        "June", "July", "August", "September", "October", "November", "December"};  
    //declare int array og dets værdier  
    int[] day_count = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  
    //print stringarray index samt intarray index  
    System.out.println(months[index-1] + " has " + day_count[index-1] + " days!");  
}  
}
```

Array typer:

int-Array: Alle elementer i arrayet starter med værdien 0.

String-Array: Alle elementer i arrayet starter med værdien null

boolean-Array: Alle elementer starter med værdien false

Iteration over et String-Array:

Her printes alle dele i det array, indtil den sidste string

```
String[] stringArray = {"bla", "bla", "bla"};

for (int i=0; i < stringArray.length; i++)
{
    System.out.println(stringArray[i])
}
//blablabla
```

Array med dynamisk variabel:

```
public static void main(String[] args) {
    int[] monthsNormal = {31,28,31,30,31,30,31,31,30,31,30,31};
    int[] monthsLeap = {31,29,31,30,31,30,31,31,30,31,30,31};

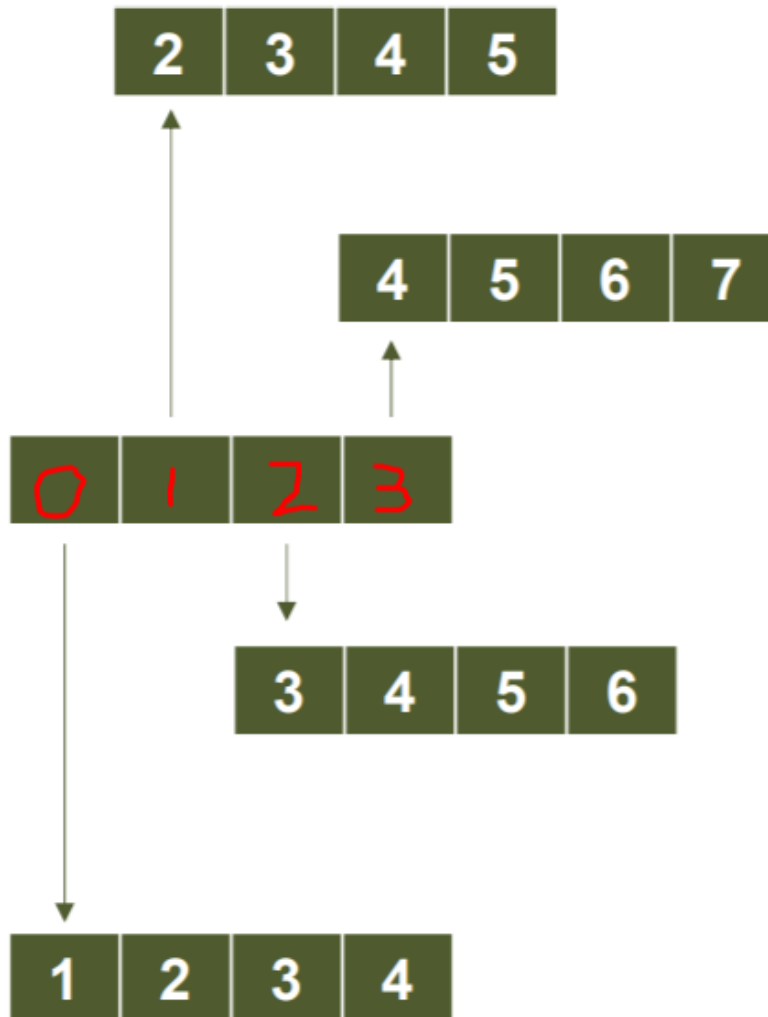
    for (int i=0; i<2020; i++)
    {
        int[] months = (i%4==0 ? monthsLeap : monthsNormal);
        System.out.println("in year " +i+ "February is " +months[1] + "days long");
    }

}
```

Array af array (Jagged array):

Man kan godt lave et array af et array:

```
int [][] as =
{
    //Alle arrays inde i et array behøver ikke at være lige store
    {1,2,3,4},
    {2,3,4,5},
    {3,4,5,6},
    {4,5,6,7}
};
System.out.println(Arrays.toString(as[2])); //printer array på plads 2; 3,4,5,6
System.out.println(as[2][3]); //printer tallet på plads 3 i array 2; 6
```



▼ Opgave

7.1 Hvad er et array?

En datastruktur, hvor alle elementer har den samme datatype

7.2 Anvendelse

Hvis man skal declare og tildele værdier til en masse statements, kan man med fordel lave et array, der gør det for en

7.3 Type af indhold vs Type af array

Et integer array kan kun indeholde hele tal, og et string array kan kun indeholde strings

7.5 Deklaration af størrelse

Eks.

```
//Tallet inde i den firkantede parentes definere længden af arrayet  
Int [12]
```

7.10 Længden af en måned

Skriv et program, der givet en måneds nummer udskriver antallet af dage i denne måned (se bort fra skudår).

```
public static void main(String[] args) {  
    int index = 5;  
    String[] months = { "January", "February", "March", "April",  
        "May", "June", "July", "August", "September", "October", "November", "December"};  
    int[] day_count = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  
    System.out.println(months[index-1] + " has " + day_count[index-1] + " days!");  
}  
}
```

7.7 Sudoku Plade

Hvordan ville du repræsentere en sudoku plade i Java?

Hvordan hænger denne datastruktur sammen i hukommelsen?

Nedenfor vises en sudoku plade i et array:

Når man gerne vil finde en værdi, skriver man eks. ([as[2][2]), hvilket er en reference til det tredje array og den tredje plads på det array. (*tredje* fordi det er 0-indekseret)

Hvis man kopierer arrays, kan man få et **Shallow copy**

```
public static void main(String[] args)  
{  
    int[][] as =  
    {  
        //Alle arrays inde i et array behøver ikke at være lige store  
        {1, 2, 3, 4, 5, 6, 7, 8, 9},  
        {4, 5, 6, 7, 8, 9, 1, 2, 3},  
        {7, 8, 9, 1, 2, 3, 4, 5, 6},  
        {2, 3, 4, 5, 6, 7, 8, 9, 1},  
        {5, 6, 7, 8, 9, 1, 2, 3, 4},  
        {8, 9, 1, 2, 3, 4, 5, 6, 7},  
        {3, 4, 5, 6, 7, 8, 9, 1, 2},  
        {6, 7, 8, 9, 1, 2, 3, 4, 5},  
    }  
}
```

```

        {9, 1, 2, 3, 4, 5, 6, 7, 8}
    };
    System.out.println(as[2][2]);

```

7.8 Areal af cirkler

Skriv et program der udregner og udskriver arealet ($\pi \cdot r^2$) af tre forskellige cirkler med radius 1, 3, og 5

```

public class string
{
    public static void main(String[] args)
    {
        double[] radius = {1, 2, 3};

        for (int i=0; i < radius.length; i++)
        {
            System.out.println(radius[i] * radius[i] * 3.14159);
        }
    }
}

```

7.12 Kalender

Skriv et program, hvori

- 1. En variabel initialiseres til at være et array der indeholder antallet af dage i hver af de 12 måneder i et normalt år.**
- 2. En anden variabel initialiseres til at være et array der indeholder antallet af dage i hver af de 12 måneder i et skudår.**
- 3. Gennemløb årene 2000 til 2020.**
- 4. Brug en ny variabel af samme type som de to forregående til at holde styr på det aktuelt korrekte array.**
- 5. For hvert år udskrives indholdet af det array den tredje variabel peger på.**

```

public class string
{
    public static void main(String[] args)
    {
        int[] monthNormal = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        int[] monthLeap = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

        for(int i = 2000; i < 2021; i++)
        {
            int[] months = (i%4 == 0 ? monthLeap : monthNormal);
            System.out.println("In year " + i + " February has " + months[1] + " days");
        }
    }
}

```

7.14 Sudoku Checker

Skriv et program, hvori

- En variabel initialiseres til at indeholde en udfyldt sudokuplade.
- Skriv kode der vurderer om pladen repræsenterer en korrekt løsning.
- Udskriv resultatet af denne vurdering.

```
public class sudoku
{
    public static int[][] sudokuMap =
    {
        {1, 2, 3, 4, 5, 6, 7, 8, 9},
        {4, 5, 6, 7, 8, 9, 1, 2, 3},
        {7, 8, 9, 1, 2, 3, 4, 5, 6},
        {2, 3, 4, 5, 6, 7, 8, 9, 1},
        {5, 6, 7, 8, 9, 1, 2, 3, 4},
        {8, 9, 1, 2, 3, 4, 5, 6, 7},
        {3, 4, 5, 6, 7, 8, 9, 1, 2},
        {6, 7, 8, 9, 1, 2, 3, 4, 5},
        {9, 1, 2, 3, 4, 5, 6, 7, 8}
    };

    static boolean verifyRow(int row)
    {
        int[] test = {0,0,0,0,0,0,0,0,0};
        for (int col = 0; col < 9; col++)
        {
            test[sudokuMap[row][col] - 1]++;
        }
        for (int i = 0; i < 9; i++)
        {
            if (test[i] != 1) {
                return false;
            }
        }
        return true;
    }

    static boolean verifyRows()
    {
        for(int row = 0; row < 9; row++)
        {
            if(!verifyRow(row))
            {
                System.out.println("Row " + row + " is invalid ");
                return false;
            }
        }
        return true;
    }

    static boolean verifyCol(int col)
    {
        int[] test = {0,0,0,0,0,0,0,0,0};
        for (int row = 0; row < 9; row++)
        {
            test[sudokuMap[row][col] - 1]++;
        }
    }
}
```

```

        test[sudokuMap[row][col] - 1]++;
    }
    for (int i = 0; i < 9; i++)
    {
        if (test[i] != 1)
        {
            return false;
        }
    }
    return true;
}
static boolean verifyCols()
{
    for(int col = 0; col < 9; col++)
    {
        if(!verifyCol(col))
        {
            System.out.println("Col " + col + " is invalid ");
            return false;
        }
    }
    return true;
}
static public boolean verifyBox (int m, int n)
{
    int [] test = {0,0,0,0,0,0,0,0,0};
    for (int row = m*3; row < (m*3)+3; row++)
    {
        for(int col = n*3; col < (n*3)+3; col++)
        {
            test[sudokuMap[row][col]-1]++;
        }
    }

    for (int i = 0; i < 9; i++)
    {
        if (test[i] != 1)
        {
            return false;
        }
    }
    return true;
}
static public boolean verifyBoxes ()
{
    for (int m = 0; m < 3; m++)
    {
        for(int n = 0; n < 3; n++)
        {
            if (!verifyBox(m,n))
            {
                System.out.println("Box [" + m + "][" + n + "] is invalid!");
                return false;
            }
        }
    }
    return true;
}
static public boolean verify()

```

```

    {
        if( verifyCols() && verifyRows() && verifyBoxes())
            return true;
        /*hvis der kun er én ting efter if statement, kan man undlade brackets
        og så hopper den bare til næste linje*/
        return false;

    }
    public static void main(String[] args)
    {
        if(verify())
            System.out.println("Sudoku is valid!");
        else
            System.out.println("Sudoku is invalid!");
    }
}

```