

PSEUDO KODE

Programming E25

ALGORITME

Definition

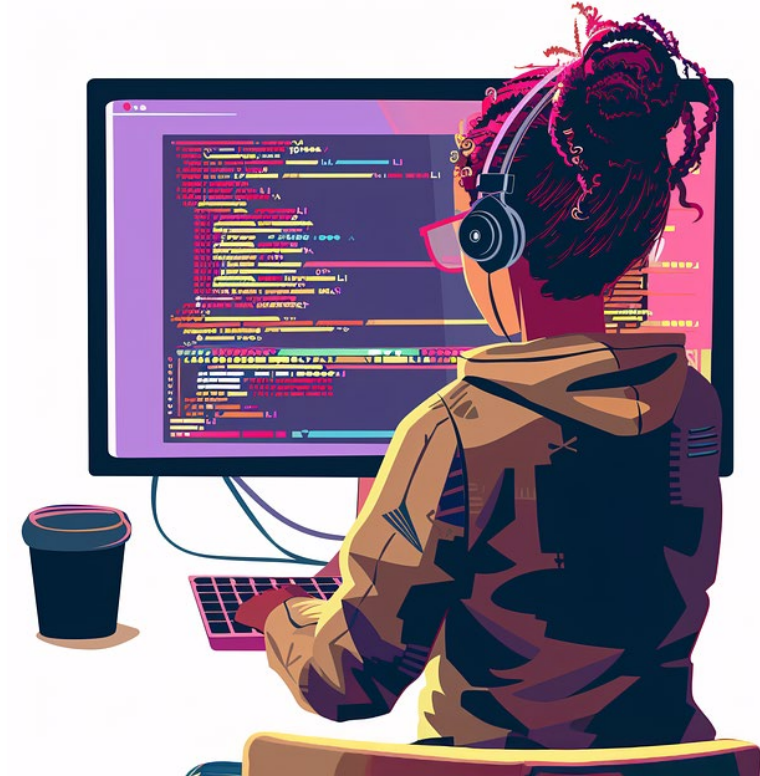
En algoritme er en rækkefølge af entydige, eksekverbare trin, der definerer en afsluttende proces

Eksempler

Madopskrifter

Strikkeopskrifter

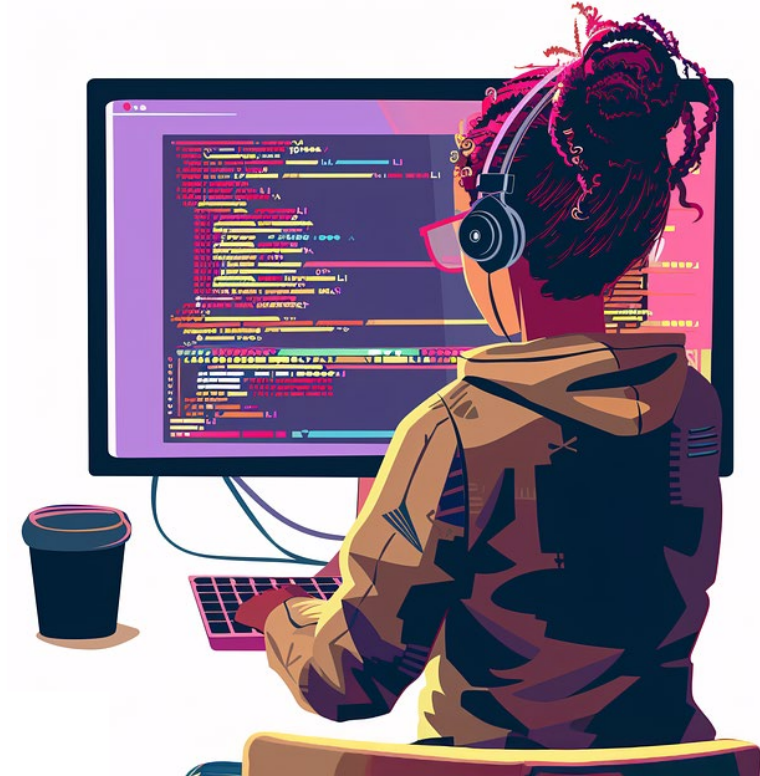
Rutevejledninger



PSEUDOKODE

Definition

En pseudokode er en metode til at gennemtænke og præcisere en algoritme før selve programmeringen

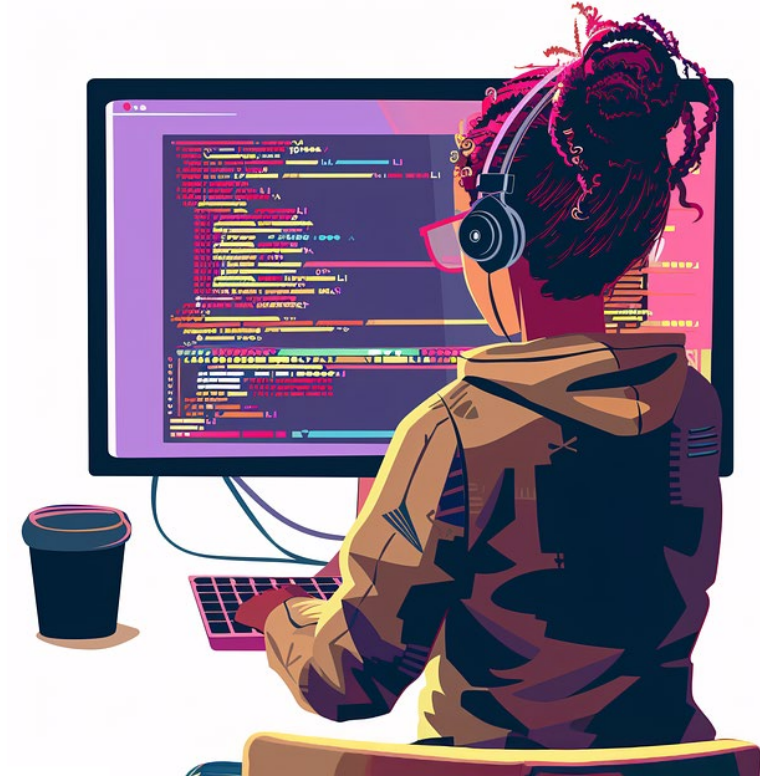


PSEUDOKODE

Definition

En pseudokode er en metode til at gennemtænke og præcisere en algoritme før selve programmeringen

- korte instruktioner i dagligdags sprog
- letlæselig for mennesker
- springbræt til programmeringen



PSEUDOKODE

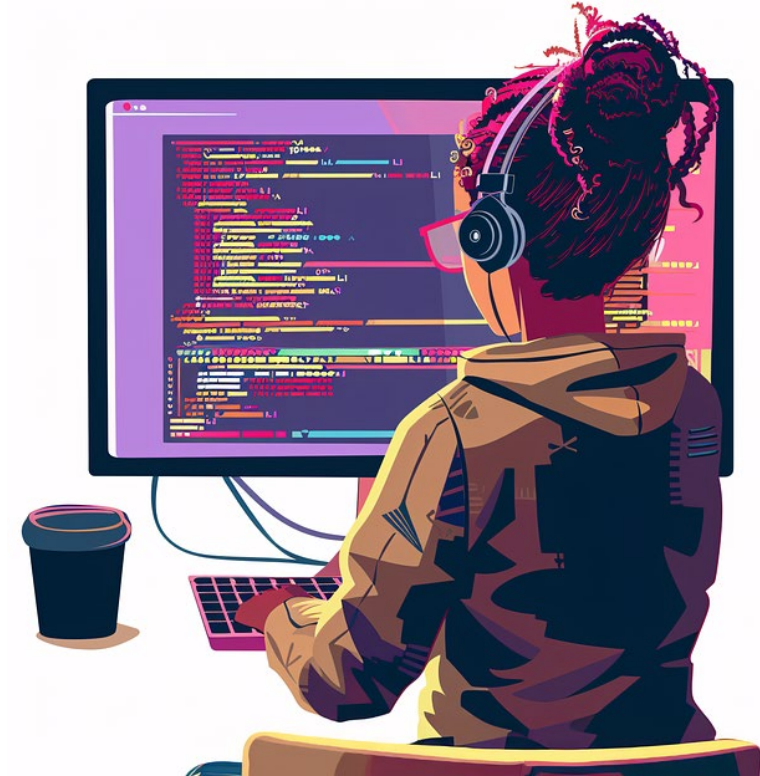
Definition

En pseudokode er en metode til at gennemtænke og præcisere en algoritme før selve programmeringen

- korte instruktioner i dagligdags sprog
- letlæselig for mennesker
- springbræt til programmeringen

Pseudokoder kan i praksis være udformet på mange forskellige måder

I denne præsentation vil jeg fremlægge én måde, som jeg finder anvendelig



SKRIVEBORDSTEST

Definition



SKRIVEBORDSTEST

Definition

En skrivebordstest er en test af en algoritme i form af en pseudokode eller et computerprogram, hvor man på 'papiret' tester de enkelte trin for at sikre sig, at algoritmen fungerer efter hensigten



SKRIVEBORDSTEST

Definition

En skrivebordstest er en test af en algoritme i form af en pseudokode eller et computerprogram, hvor man på 'papiret' tester de enkelte trin for at sikre sig, at algoritmen fungerer efter hensigten

Bemærk, at en skrivebordstest er en form for unplugged programmering, idet en skrivebordstest kan udformes som et rollespil med flere elever, hvor hver elev har fået en rolle, fx en bestemt variabel



PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

hvis tal1 er større end tal2

udskriv, at tal1 er størst

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

hvis tal1 er større end tal2

udskriv, at tal1 er størst

ellers hvis tal2 er større end tal1

udskriv, at tal2 er størst

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

hvis tal1 er større end tal2

udskriv, at tal1 er størst

ellers hvis tal2 er større end tal1

udskriv, at tal2 er størst

ellers

udskriv, at de er lige store

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

hvis tal1 er større end tal2

udskriv , at tal1 er størst

ellers hvis tal2 er større end tal1

udskriv , at tal2 er størst

ellers

udskriv , at de er lige store

nøgleord

PSEUDOKODE

En pseudokode, der bestemmer det største af to tal

indlæs det ene tal

sæt tal1 lig med det indlæste tal

indlæs det andet tal

sæt tal2 lig med det indlæste tal

hvis tal1 er større end tal2

→ udskriv , at tal1 er størst

ellers hvis tal2 er større end tal1

→ udskriv , at tal2 er størst

ellers

udskriv , at de er lige store

nøgleord

bemærk indrykninger
(= indentations)

PSEUDOKODE

tildelinger (= assignments)

forgreninger (= branchings)

gentagelser (= loops)

udtryk (= expressions)

betingelser (= conditions)

input (= inddata)

output (= uddata)

funktioner (= functions)



TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

sæt navn lig med Peter

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

sæt navn lig med Peter

læg score til den totale score

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

sæt navn lig med Peter

læg score til den totale score

tilføj 02 til listen af karakterer

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

sæt navn lig med Peter

læg score til den totale score

tilføj 02 til listen af karakterer

beregn gennemsnittet

TILDELINGER (= ASSIGNMENTS)

Tildelinger bruges, når variable får værdier

Eksempler

sæt score lig med 0

sæt karakterer lig med 7, 4, 4, 10, 12 og 7

sæt navn lig med Peter

læg score til den totale score

tilføj 02 til listen af karakterer

beregn gennemsnittet

registrer, at tallet er fundet

FORGRENINGER (= BRANCHINGS)

Forgreninger bruges, når det videre forløb afhænger af en betingelse

FORGRENINGER (= BRANCHINGS)

Forgreninger bruges, når det videre forløb afhænger af en betingelse

Eksempler

hvis din score er større end min
udskriv, at du vandt

FORGRENINGER (= BRANCHINGS)

Forgreninger bruges, når det videre forløb afhænger af en betingelse

Eksempler

hvis din score er større end min
udskriv, at du vandt

hvis mønten giver plat
udskriv, at du bestemmer

ellers
udskriv, at jeg bestemmer

FORGRENINGER (= BRANCHINGS)

Forgreninger bruges, når det videre forløb afhænger af en betingelse

Eksempler

hvis din score er større end min

→ udskriv, at du vandt

hvis mønten giver plat

→ udskriv, at du bestemmer

ellers

→ udskriv, at jeg bestemmer

bemærk indrykninger
(= indentations)

GENTAGELSER (= LOOPS)

Gentagelser bruges, når en eller flere hændelser skal gentages

GENTAGELSER (= LOOPS)

Gentagelser bruges, når en eller flere hændelser skal gentages

Eksempler

for hvert tal i listen

læg tal til summen

GENTAGELSER (= LOOPS)

Gentagelser bruges, når en eller flere hændelser skal gentages

Eksempler

for hvert tal i listen

læg tal til summen

så længe der er flere varer i indkøbskurven

indtast prisen på den næste vare

læg prisen til den samlede pris

GENTAGELSER (= LOOPS)

Gentagelser bruges, når en eller flere hændelser skal gentages

Eksempler

- for hvert tal i listen

 - læg tal til summen

- så længe der er flere varer i indkøbskurven

 - indtast prisen på den næste vare

 - læg prisen til den samlede pris

- gør 4 gange

 - tegn en linje med længde 5

 - drej 90 grader

GENTAGELSER (= LOOPS)

Gentagelser bruges, når en eller flere hændelser skal gentages

Eksempler

for hvert tal i listen

→ læg tal til summen

så længe der er flere varer i indkøbskurven

→ indtast prisen på den næste vare

→ læg prisen til den samlede pris

gør 4 gange

→ tegn en linje med længde 5

→ drej 90 grader

bemærk indrykninger
(= indentations)

UDTRYK (= EXPRESSIONS)

Eksempler

summen af 2 og 3

(et tal)

UDTRYK (= EXPRESSIONS)

Eksempler

summen af 2 og 3

(et tal)

peter efterfulgt af andersen

(en streng)

UDTRYK (= EXPRESSIONS)

Eksempler

summen af 2 og 3

(et tal)

peter efterfulgt af andersen

(en streng)

min vægt er større end din vægt

(en sandhedsværdi)

UDTRYK (= EXPRESSIONS)

Eksempler

summen af 2 og 3

(et tal)

peter efterfulgt af andersen

(en streng)

min vægt er større end din vægt

(en sandhedsværdi)

Bemærk, at udtryk i en pseudokode skrives uden matematiske symboler

+, -, <, >, <= og så videre

da brugen af matematiske symboler ikke nødvendigvis er ens i forskellige programmeringssprog

BETINGELSER (= CONDITIONS)

Betingelser er udtryk, der giver en sandhedsværdi, og som bruges i forgreninger og gentagelser

BETINGELSER (= CONDITIONS)

Betingelser er udtryk, der giver en sandhedsværdi, og som bruges i forgreninger og gentagelser

Eksempler

min vægt er større end din vægt

BETINGELSER (= CONDITIONS)

Betingelser er udtryk, der giver en sandhedsværdi, og som bruges i forgreninger og gentagelser

Eksempler

min vægt er større end din vægt

konditallet ligger mellem 40 og 50

BETINGELSER (= CONDITIONS)

Betingelser er udtryk, der giver en sandhedsværdi, og som bruges i forgreninger og gentagelser

Eksempler

min vægt er større end din vægt

konditallet ligger mellem 40 og 50

navnet er skrevet med små bogstaver

BETINGELSER (= CONDITIONS)

Betingelser er udtryk, der giver en sandhedsværdi, og som bruges i forgreninger og gentagelser

Eksempler

min vægt er større end din vægt

konditallet ligger mellem 40 og 50

navnet er skrevet med små bogstaver

Bemærk, at udtryk i en pseudokode skrives uden matematiske symboler

+, -, <, >, <= og så videre

da brugen af matematiske symboler ikke nødvendigvis er ens i forskellige programmeringssprog

INPUT (= INDDATA)

Input bruges til at indlæse data fra brugeren

INPUT (= INDDATA)

Input bruges til at indlæse data fra brugeren

Eksempler

indtast din alder

INPUT (= INDDATA)

Input bruges til at indlæse data fra brugeren

Eksempler

indtast din alder

indtast, om du ønsker at fortsætte med at...

INPUT (= INDDATA)

Input bruges til at indlæse data fra brugeren

Eksempler

indtast din alder

indtast, om du ønsker at fortsætte med at...

indtast næste karakter

OUTPUT (= UDDATA)

Eksempler

udskriv karaktergennemsnittet

OUTPUT (= UDDATA)

Eksempler

udskriv karaktergennemsnittet

udskriv den sorterede rækkefølge af ...

OUTPUT (= UDDATA)

Eksempler

udskriv karaktergennemsnittet

udskriv den sorterede rækkefølge af ...

tegn huset

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

gør 4 gange

tegn en linje med længde 5

drej 90 grader

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

gør 4 gange

tegn en linje med længde 5

drej 90 grader

tegn et kvadrat med sidelængde s

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

gør 4 gange

tegn en linje med længde 5

drej 90 grader

tegn et kvadrat med sidelængde s

gør 4 gange

tegn en linje med længde s

drej 90 grader

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

gør 4 gange

tegn en linje med længde 5

drej 90 grader

tegn et kvadrat med sidelængde

gør 4 gange

tegn en linje med længde s

drej 90 grader

s

input til funktionen
(parameter)

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

tegn et kvadrat med sidelængde 5

- gør 4 gange
 - tegn en linje med længde 5
 - drej 90 grader

tegn et kvadrat med sidelængde s

- gør 4 gange
 - tegn en linje med længde s
 - drej 90 grader

bemærk indrykninger
(= indentations)

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

beregn gennemsnit af en række tal

sæt summen lig med 0

sæt antal lig med 0

så længe der er flere tal

indlæs næste tal

læg tallet til summen

læg 1 til antal

sæt snit til summen delt med antal

returner snit

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

beregn gennemsnit af en række tal

- sæt summen lig med 0
- sæt antal lig med 0
- sålænge der er flere tal
 - indlæs næste tal
 - læg tallet til summen
 - læg 1 til antal
- sæt snit til summen delt med antal
- returner snit

bemærk indrykninger
(= indentations)

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

beregn gennemsnit af

en række tal

input til funktionen
(parameter)

- sæt summen lig med 0
- sæt antal lig med 0
- så længe der er flere tal
 - indlæs næste tal
 - læg tallet til summen
 - læg 1 til antal
- sæt snit til summen delt med antal
- returner snit

bemærk indrykninger
(= indentations)

FUNKTIONER (= FUNCTIONS)

Funktioner er en blok af pseudokoder med en specifik opgave

beregn gennemsnit af

en række tal

input til funktionen
(parameter)

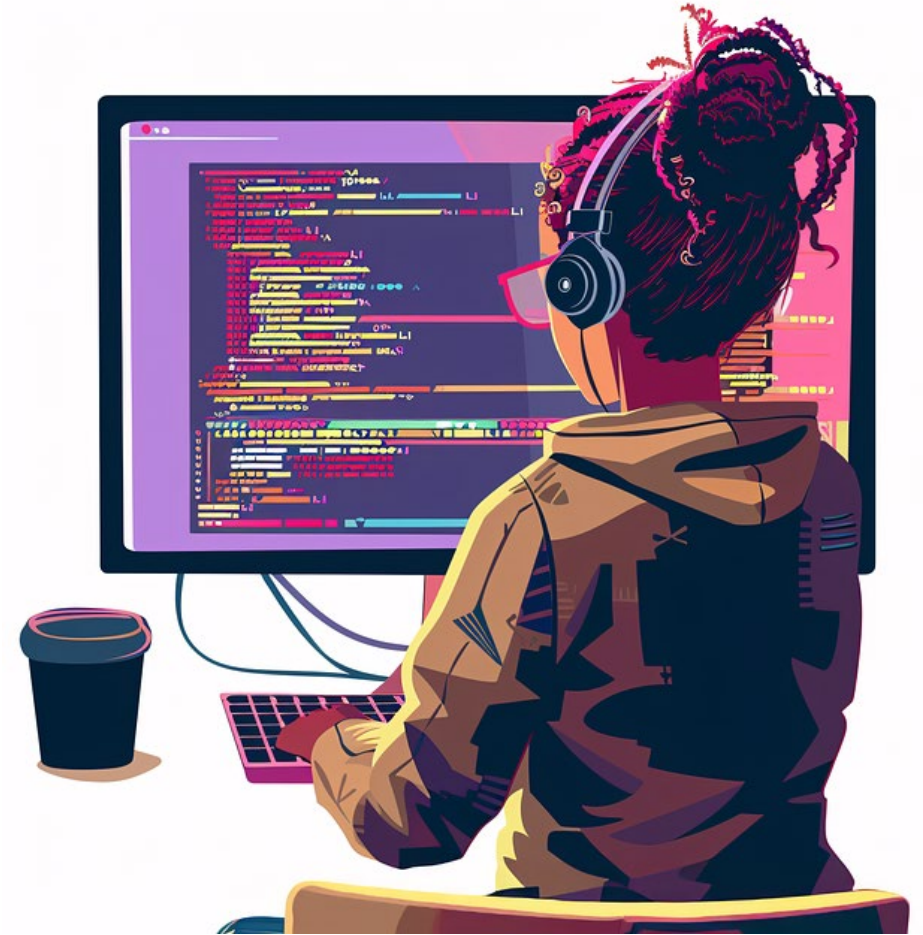
- sæt summen lig med 0
- sæt antal lig med 0
- så længe der er flere tal
 - indlæs næste tal
 - læg tallet til summen
 - læg 1 til antal
- sæt snit til summen delt med antal
- returner snit

output

bemærk indrykninger
(= indentations)

FORDELE VED PSEUDOKODE

—
Lettere at læse og forstå algoritmen



FORDELE VED PSEUDOKODE

—
Lettere at læse og forstå algoritmen

Simplere at lave algoritmen



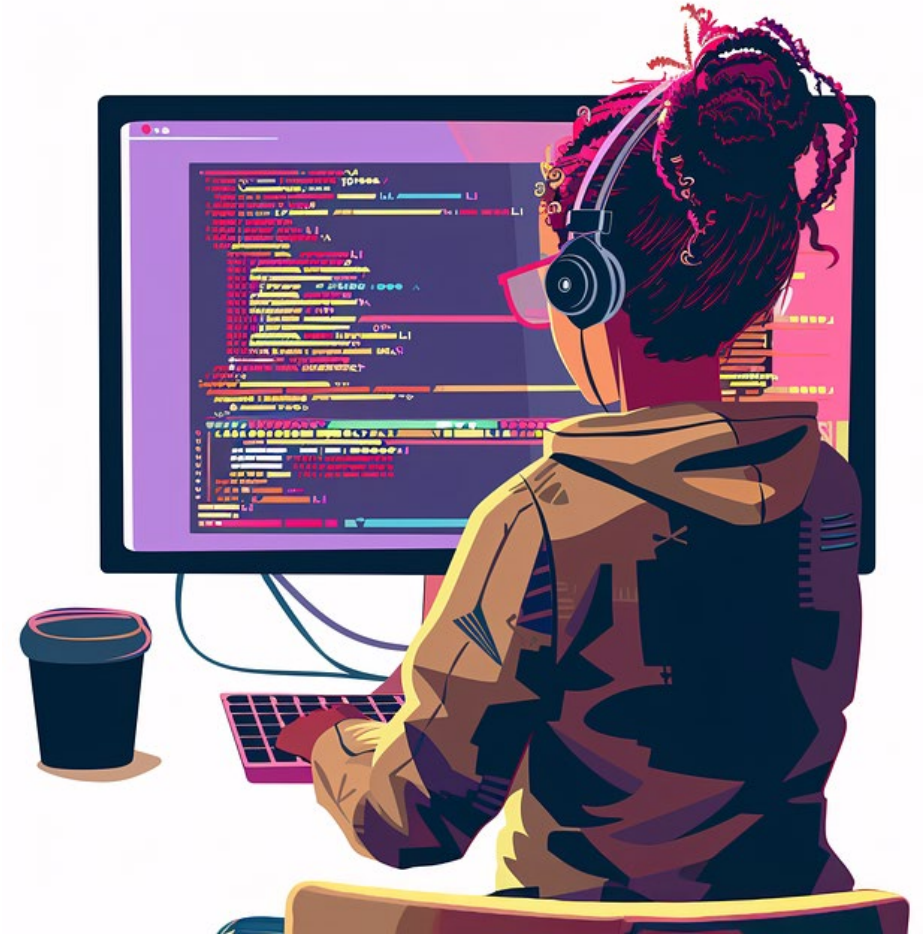
FORDELE VED PSEUDOKODE

—

Lettere at læse og forstå algoritmen

Simplere at lave algoritmen

Lettere at udvide algoritmen



FORDELE VED PSEUDOKODE

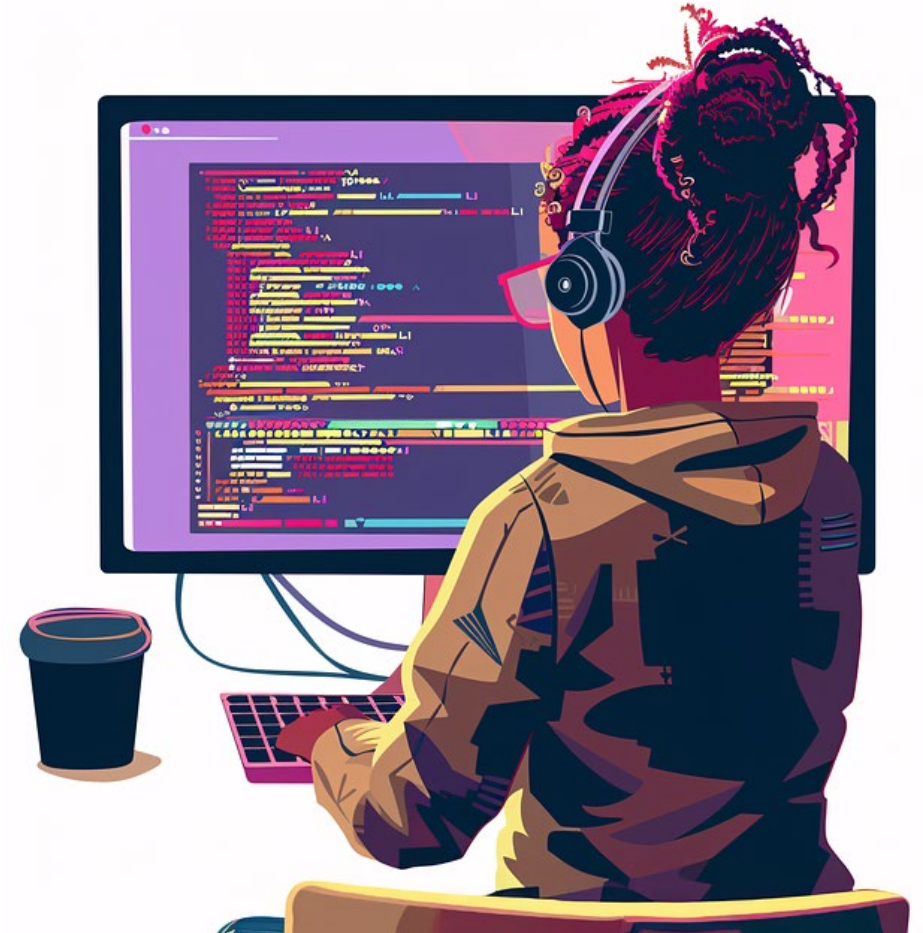
—

Lettere at læse og forstå algoritmen

Simplere at lave algoritmen

Lettere at udvide algoritmen

Fokus på algoritmisk tænkning



FORDELE VED PSEUDOKODE

—

Lettere at læse og forstå algoritmen

Simplere at lave algoritmen

Lettere at udvide algoritmen

Fokus på algoritmisk tænkning

Pseudokode er universel



FORDELE VED PSEUDOKODE

—

- Lettere at læse og forstå algoritmen

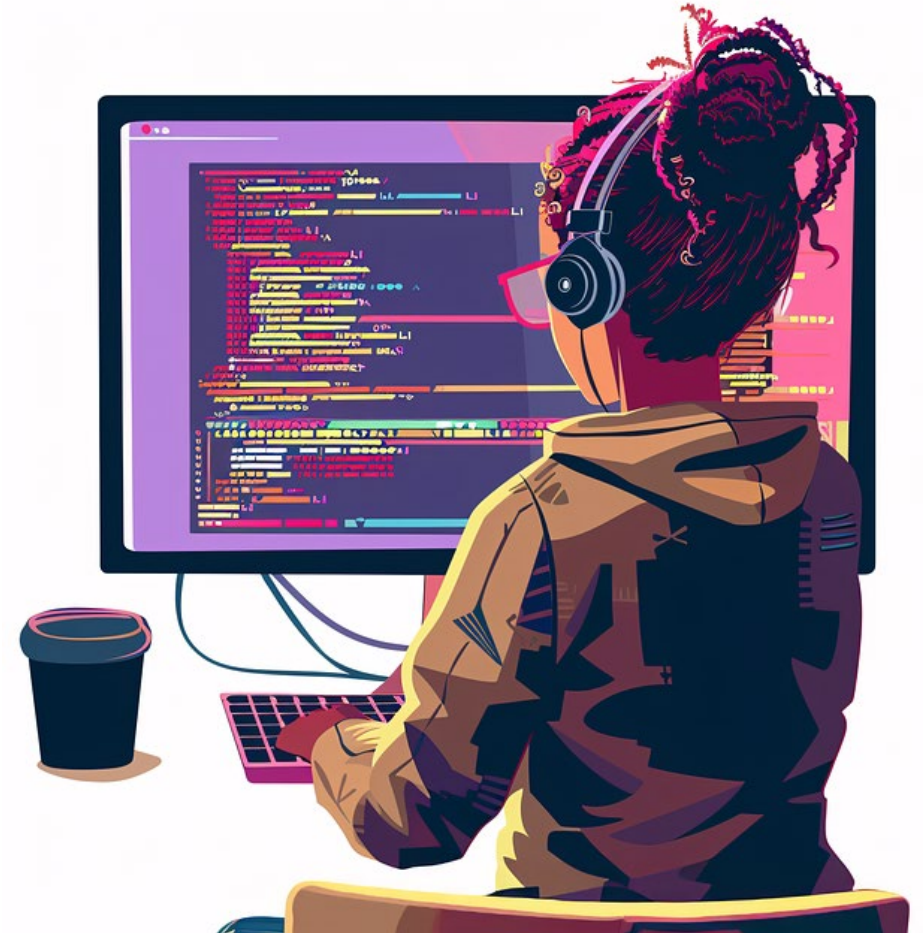
- Simplere at lave algoritmen

- Lettere at udvide algoritmen

- Fokus på algoritmisk tænkning

- Pseudokode er universel

- Springbræt til alle programmeringssprog



FORDELE VED PSEUDOKODE

Lettere at læse og forstå algoritmen

Simplere at lave algoritmen

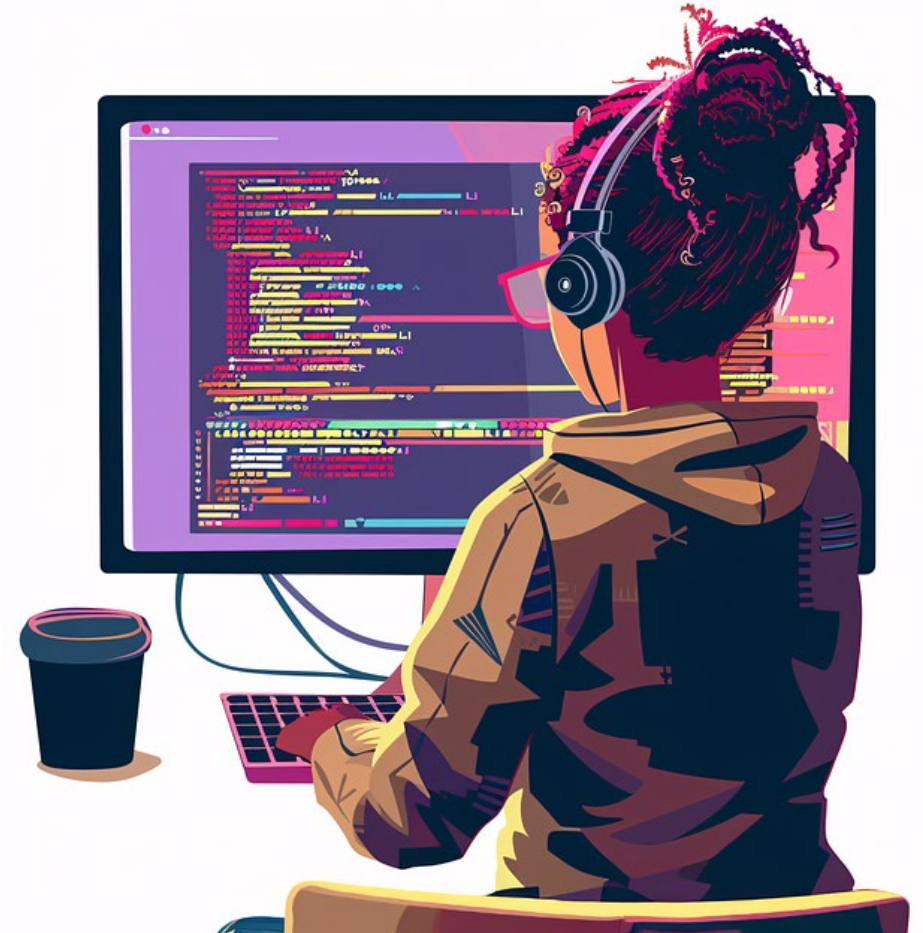
Lettere at udvide algoritmen

Fokus på algoritmisk tænkning

Pseudokode er universel

Springbræt til alle programmeringssprog

Fokus på semantik fremfor syntaks



FORDELE VED PSEUDOKODE

—

- Lettere at læse og forstå algoritmen

- Simplere at lave algoritmen

- Lettere at udvide algoritmen

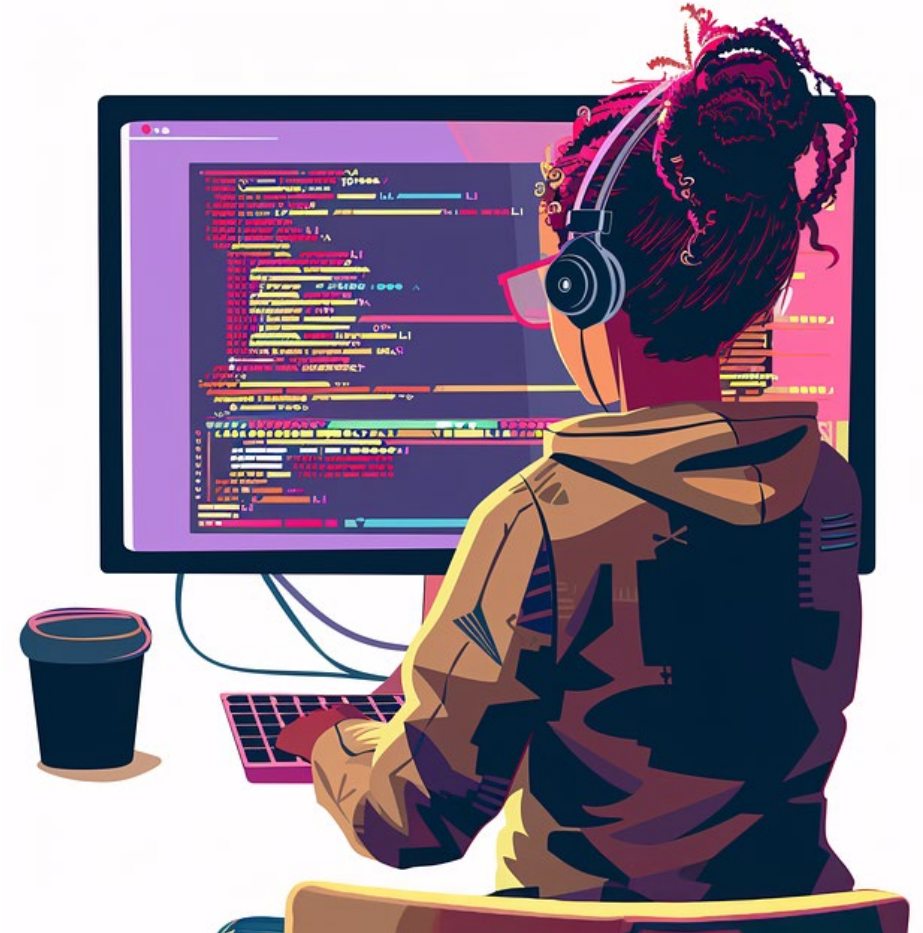
- Fokus på algoritmisk tænkning

- Pseudokode er universel

- Springbræt til alle programmeringssprog

- Fokus på semantik fremfor syntaks

- Giver god dokumentation til programmet



FORDELE VED PSEUDOKODE

-
- Lettere at læse og forstå algoritmen
- Simplere at lave algoritmen
- Lettere at udvide algoritmen
- Fokus på algoritmisk tænkning
- Pseudokode er universel
- Springbræt til alle programmeringssprog
- Fokus på semantik fremfor syntaks
- Giver god dokumentation til programmet
- Nemmere at rette fejl (med skrivebordstjek)

