**Oppgave 1**

Hvilket av følgende utsagn er ikke korrekt i objektorientert programmering? (Kun ett svar.)

1. Konstruktøren \_\_init\_\_ er en metode hvor den første parameteren alltid vil vise til objektet selv.
2. Metoder sammenlignes med funksjoner, men må alltid ha minst en parameter for å virke.
3. Objekter er variabler, mens klassen er oppskriften på hvordan objektet lages.
4. Når Klasse2 lages ved at den arver fra Klasse1 så vil Klasse1 inneholde alle attributtene og metodene til Klasse2.

Svar: d

**Oppgave 2**

Gitt klassen Sirkel med fire parametre i konstruktøren: self, xpos, ypos og radius, hvor bare parameteren radius har en defaultverdi, hvilke(n) av disse programlinjene gir alltid en feilmelding i python? (Det kan være flere svar.)

1. s1 = Sirkel(1, 2)
2. s2 = Sirkel(3, 4, 5)
3. s3 = Sirkel(self,6,7,8)
4. s4 = Sirkel(9,radius=10)
5. s5 = Sirkel(xpos=11, ypos=12, radius=13)
6. s6 = Sirkel(radius=14, xpos=15, ypos=16)
7. s7 = Sirkel(radius=17, 18, 19)
8. s8 = Sirkel("a","b","c")

Svar: c, d, g

**Oppgave 3**

Velg mellom linjene med pseudokode for å skrive et program.

Programmet skal ha en funksjon som regner ut differansen mellom to tall.

Programmet skal bruke funksjonen til å regne ut 4 minus 5

Output fra programmet skal være -1.

Funksjonen skal kunne brukes om igjen i et annet program hvor variablene a og b ikke finnes.

Husk å lage innrykk for kode inne i en blokk.

CALL diff

DISPLAY a - b

DISPLAY CALL diff(a,b)

ENDFUNCTION

FUNCTION diff

FUNCTION diff(a,b)

RETURN a - b

SET a TO 4

SET b TO 5

Svar (kopier eller flytt de linjene du vil ha med):

FUNCTION diff(a,b)

RETURN a - b

ENDFUNCTION

SET a TO 4

SET b TO 5

DISPLAY CALL diff(a,b)

**Oppgave 4**

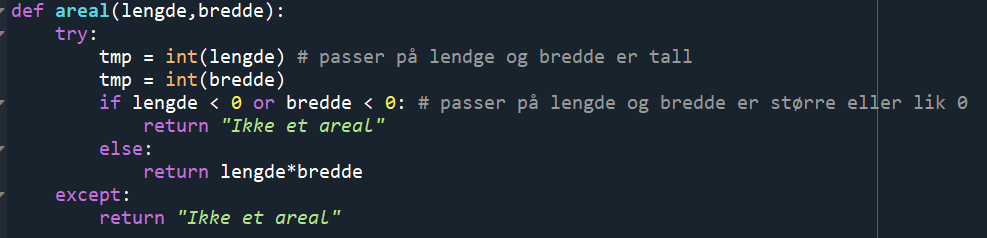
|  |
| --- |
| def areal(lengde,bredde):      return lengde\*bredde    def test\_areal():        verdier\_inn = [-1000000000,-1,0,0.5,1,10,1000000000]        arealer\_forventet = ["Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal","Ikke et areal",0,0.0,0,0,0,"Ikke et areal","Ikke et areal",0.0,0.25,0.5,5.0,500000000.0,"Ikke et areal","Ikke et areal",0,0.5,1,10,1000000000,"Ikke et areal","Ikke et areal",0,5.0,10,100,10000000000,"Ikke et areal","Ikke et areal",0,500000000.0,1000000000,10000000000,1000000000000000000]        i = 0      for lengde in verdier\_inn:          for bredde in verdier\_inn:              test = areal(lengde,bredde)              kontroll = arealer\_forventet[i]              if test == kontroll:                  print("Funksjonen virker")              else:                  print(f"Funksjonen areal {test:25.2f} virker ikke for lengden {lengde:15.1f} og bredden {bredde:15.1f}")              i += 1    test\_areal() |

1. Beskriv kort hva funksjonen test\_areal gjør. Forklar hvorfor man kjører slike tester og foreslå et par verdier til som burde være med i verdier\_inn hvis listen skal kunne brukes generelt til å teste forskjellige programmer som behandler tall. (Tips: programmet er vedlagt som python-fil.)

Svar: test\_areal() er en funksjon som tester funksjonen areal(). Den passer på at resultatet av areal() funksjonen er det som er forventet med parameterne som er satt inn, testing som dette forhindrer at funksjoner skaper problemer med resten av koden i programmet. Test\_areal() funksjonen har en liste av verdier med forskjellige typer parametere men for å forbedre det burde det inkludere andre datatyper som Bollean, Strings, eller chars f.eks. True, ‘’string’’ eller ‘c’.

1. Gjør endringer i funksjonen areal slik at man kun får "Funksjonen virker" når man kjører test\_areal. Funksjonen test\_areal skal ikke endres. Skriv den nye areal-funksjonen inn her:

Svar:



**Oppgave 5**

Et bilde som inneholder diagram, tekst, sketch, tegning

Automatisk generert beskrivelse

Skriv hva som kommer i output ved følgende input i programmet beskrevet i flytdiagrammet:

|  |  |  |
| --- | --- | --- |
| input a | input b | Output |
| 4 | 5 | Svar:  9.0  4  5 |
| 4.5 | 5 | Svar:  9.5  4.5  5 |
| 4,5 | 5 | Svar:  Kunne ikke summere verdiene  4,5  5 |
|  | 5 | Svar:  Kunne ikke summere verdiene  5 |

\*den fjerde verdien for a er at man ikke skriver inn noen verdi

**Oppgave 6**

Unni prøver å lære seg genetisk programmering. Hun følger et eksempel, genetic\_theBuilder.py (gitt ut på forberedelsesdagen), og lager sin egen kode basert på dette.

Som mattelærer har hun irritert seg over Geogebra sin manglende evne til å finne nullpunktet til den annenderiverte av en *logistisk funksjon*, av og til går det, men ofte går det ikke.

Unni starter med en litt enklere variant av problemstillingen: for hvilken x-verdi er den logistiske funksjonen halvveis til bæreevnen?

|  |
| --- |
| Ved en *logistisk vekst* er antall gitt ved    der C, a og b er konstanter.    Etter lang tid vil antallet nærme seg C.  Vekstfarten er størst i vendepunktet der . Antallet er da    For eksempel:  a = 20  b = 0.1  C = 50  gir x = ln(20)/0,1 29,957 og y = 50/2 = 25. |

Kan vi finne disse verdiene for x og y ved genetisk programmering? Unni gjør et forsøk:

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt    # Konstanter for en logistisk funksjon:  # f(x) = C / ( 1 + ae^-bx )  a = 20  b = 0.1  C = 50    # Konstanter for løsningen  antall\_forslag = 100 # antall løsninger vi tester parallelt  behold = 10 # antall løsninger vi beholder fra hver generasjon, max lik antall\_forslag    grense = 1000 # startverdier for x er innenfor [-grense,grense> når vi bruker np.random.uniform(fra,til,antall)  maal = 9999 # 1 / nøyaktighet    utskrift = 1 # antall linjer skrevet pr generasjon  generasjoner = 1000 # antall runder med mutasjoner før vi gir oss, hvis vi ikke får ønsket nøyaktighet    def f(x,a,b,C):      """      Logistisk funksjon      """      ebx = np.e\*\*(-b\*x)      return C/(1+a\*ebx)    # Sjekk grafen for funksjonen  xverdier = np.linspace(-grense,grense,1001)    def fig(funk,tittel):      plt.grid(True)      plt.axhline(y=0, xmin=0, xmax=1,color="black")      plt.axhline(y=C/2, xmin=0, xmax=1,color="blue")      plt.axvline(x=0, ymin=0, ymax=1,color="black")      plt.axvline(x=np.log(a)/b, ymin=0, ymax=1,color="blue")      plt.title(tittel)      plt.plot(xverdier,funk,"r")      plt.show()    #fig(f(xverdier,a,b,C),"Funksjonen")    # Hvor god er foreslått løsning?  def finnXfraY(x,a,b,C):      ans = f(x,a,b,C) - C/2      if ans == 0:          return 999999      else:          return float(abs(1/ans))    # Gjett y-verdier  forslag = np.random.uniform(-grense,grense,antall\_forslag)    # Finn beste løsning  for i in range(generasjoner):      sorterte\_forslag = []      for x in forslag:          passer = finnXfraY(x,a,b,C)          sorterte\_forslag.append((passer,x))        sorterte\_forslag.sort()      sorterte\_forslag.reverse()        #for j in range(utskrift):      #    print(sorterte\_forslag[j])        if sorterte\_forslag[0][0] >= maal:          break        beste = []      for k in range(behold):          beste.append(sorterte\_forslag[k][1])    # Skriv ut beste løsning  print(i,round(sorterte\_forslag[0][1],3), round(f(sorterte\_forslag[0][1],a,b,C),3)) |

1. Programmet logistisk\_vekst.py gitt over virker ikke. Forklar hva som er feil eller mangler. (Tips: programmet er vedlagt som python-fil.)

Svar: de beste resultatene fra generasjonene blir ikke brukt i neste generasjon derfor vil algoritmen aldri tilnærme seg svaret fordi for hver generasjon bruker den noen nye tilfeldige tall.

1. Gjør endringer som skal til for å finne x 29,957 og y 25,000 ved genetisk programmering.

Svaret leveres som en programfil.

**Oppgave 7**

Med referanse til artikkelen "Algoritmer styrer din hverdag", youtube-videoen "Genetic Algorithm In Python Super Basic Example" (ikke tilgjengelig under prøveeksamen) og programmet genetic\_theBuilder.py fra forberedelsesdagen, avgjør om følgende påstander er sanne eller usanne:

|  |  |
| --- | --- |
| Påstand | sant/usant |
| En algoritme er en oppskrift som forteller hvordan noe gjøres. | Svar: sant |
| Det er mer miljøvennlig å skrive effektiv kode. | Svar: sant |
| Datamaskinen Watson bistår norske sykehus med medisinske diagnoser og behandlingsforslag. | Svar: usant |
| Facebook's algoritmer går aktivt inn for å vinkle nyhetsstrømmen din med deres egen politiske agenda. | Svar: sant |
| Genetiske algoritmer etteraper biologisk evolusjon blant annet ved å sette sammen data, bytte ut data, eller mutere data, og gjenta prosessen over flere generasjoner, for å se hvilken utvikling som gir størst suksess. | Svar: sant |
| Et problem med diabetesappen som er utviklet i Norge er at man må loggføre blodsukkermålingene manuelt ved å skrive dem inn. | Svar: usant |
| Bruk av teknologi og algoritmer i pasientoppfølging har et potensiale til å redusere antall sykehusinnleggelser, dette er allerede vist i praksis. | Svar: usant  Selv om det har potensiale, er det ikke enda påvist |

**Oppgave 8**

Skriv en kort tekst om rank-algoritmer, som f.eks. brukes av Google og Facebook, men også andre leverandører. Teksten skal vurdere muligheter og utfordringer som rank-algoritmer skaper, og drøfte konsekvenser av dette for individ og samfunn.

En god lengde på teksten er omtrent 300 ord.

Svar:

**Oppgave 9**

Denne seksjonen inneholder en oppgave med to deloppgaver, hvor du skal bruke datasettet fra Materiale 4 i forberedelsesdelen.

Du bør sette av inntil 1 time til denne oppgaven.

Datasettene består av

Sysselsette: Tekststreng

Næring: Tekststreng

Inudstrirobotar\_2018: heltall

Industrirobotar\_2020: heltall

Industrirobotar\_2022: heltall

Tenesterobotar\_2018: heltall

Tenesterobotar\_2020: heltall

Tenesterobotar\_2022: heltall

csv-filen inneholder en innledende tekst (overskrift) som må fjernes før bruk. Separatoren er semikolon ";"

OBS! Det er lurt å bare legge det ene formatet av datasettet som du faktisk bruker inn i innleveringsmappen. (Du kan velge mellom JSON og CSV.) Dette for å unngå at innleveringsfilen blir for stor.

**Oppgave a:**

Lag et program som skriver ut de tre næringene med **100 eller flere ansatte** som har høyest prosentandel industriroboter i 2022. Lag også et program som skriver ut de tre næringene (i datasettet) med **100 eller flere ansatte** som har høyest prosentandel tjenesteroboter i 2022.

Fremstill dataene i et samlet eller i to separate diagrammer (f.eks. stolpediagram)

Svar leveres som program.

**Oppgave b:**

Undersøk sammenhengen mellom bedrifts-størrelse og prosentandel roboter (industriroboter eller tjenesteroboter) for tre næringer i **2020**. Vis sammenhengen for disse tre næringene og robot-typene grafisk:

* + 10-39 Industri, kraftforsyning, vatn, avløp og renovasjon (industriroboter)
  + 46 Agentur- og engroshandel med unntak av motorvogner (velg enten industri- eller tenesteroboter)
  + 58-63 Informasjon og kommunikasjon (tjenesteroboter)

x-aksen skal være antall sysselsatte: 10 til 19 sysselsatte, 20-49 sysselsatte, 50 -99 sysselsatte og 100 eller flere-kategorien. Sørg for at antall sysselsatte er i stigende rekkefølge (sortert på antall sysselsatte). Y-aksen er prosentandel roboter.

Ser det ut til å være en sammenheng mellom bedriftsstørrelse og prosentandel roboter, og i så fall hvilken sammenheng ser det ut til å være? Begrunn.

Svar: Lim inn grafer og utskrifter her og skriv inn begrunnelse. Lever programmet.

Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse

Her er det en klar sammenheng mellom antall ansatte og prosentandel av industriroboter. Når antall ansatte øker vil prosentandelen øke men ikke lineært. Selv om vi mangler data for informasjon og kommunikasjon ser vi fortsatt en økning i prosentandelen.

**Oppgave 10**

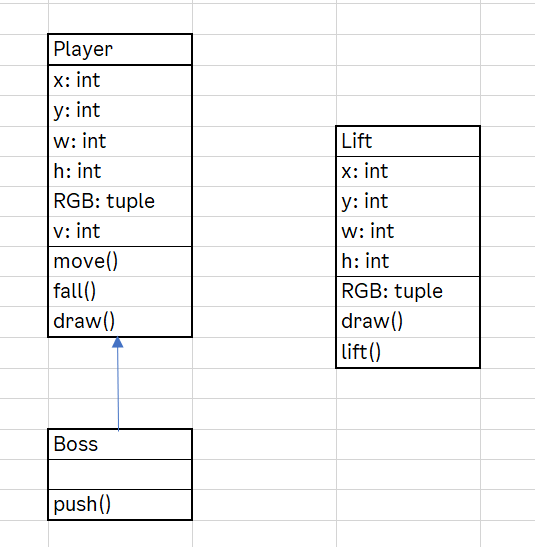
(Svarene til oppgave a og b utvikles i samspill.)

**Oppgave a:**

Lag spillet som beskrevet under, med objektorientert programmering.

**Oppgave b:**

Anvend objektorientert modellering og beskriv klassene du bruker med UML-diagrammer.



Beskrivelse og krav:

Lag et spill der en spiller (et "menneske") beveger seg i et hus med vegger, etasjer og stiger, pluss en heis (sett fra siden).

Det er en heis som tar deg til høyere nivåer (etasjer/plattformer).

Spilleren skal møte en boss. Det er ressurser tilgjengelig i rommet, som spilleren skal skaffe seg.

Spillebrettet kan være stort eller lite, f.eks. 630 x 630 piksler.

Krav til å hente «ressurser»

Spilleren kan gå frem og tilbake og kan bevege seg opp og ned langs stiger. Spilleren skal ikke kunne gå gjennom vegger eller hindringer.

Ressursene skal være enkle objekter (rektangler eller sirkler) som spilleren kan kollidere med og "ta til seg". Når du kolliderer med en ressurs fjernes den fra spillebrettet. Plasser tre ressurser rundt i huset og sett inn etasjer og stiger som spilleren kan bevege seg langs.

Ta heisen opp til "bossen"

Etter at ressursene er samlet, skal spilleren ta heisen opp til bossen som står på en etasje høyere opp (se figur under). Du velger selv om dette skjer automatisk eller om du kan styre heisen med tastaturinput. Tips: Heisen kan være tegnet med en tynn kantlinje, bruk "width = 1" i tegnemetoden for å oppnå dette (se utklipp fra pygame.draw.rect() under hvis du bruker dette)

Vinne spillet

Rekkefølgen av hendelser er:

Samle alle ressursene

Ta heisen opp til bossen

Vinn over bossen ved å (for eksempel) dytte hen ned fra platået hen står på.

Krav til mennesket:

Mennesket består av rektangulær underkropp og elliptisk hode (se eksempel på figur under).

Krav til huset

Huset skal være omsluttet av vegger, gulv og tak

Det skal være et platå i rommet, der bossen står, og det skal gå en heis opp dit.

Du setter i tillegg inn de etasjer og stiger som trengs for å samle inn ressursene.

Krav til bossen:

Bossen er samme type objekt som deg, spilleren. Han skal kunne bli dyttet ned av deg eller med annen interaksjon tape kampen mot deg.

Her ser du et eksempel på utseendet til huset underveis i utviklingen av spillet: den røde spilleren starter nederst i venstre hjørne, Bossen er den blå figuren som står ytterst på kanten av det øvre platået.

Heisen er rød og har kun et omriss, spilleren (deg) skal gå inn i heisen og kunne sette den i gang slik at både spiller og heis går opp til platået der bossen står.

Ressursene er de små kvadratene i forskjellige farger (valgfri plassering ut fra hvor du setter etasjer og stiger. Vegger og etasjer er svarte, stiger må være en annen farge.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Illustrasjon av hvordan det kan løses

|  |  |
| --- | --- |
| Et bilde som inneholder logo  Automatisk generert beskrivelseSpilleren står i startposisjonen sin | Et bilde som inneholder logo  Automatisk generert beskrivelseSpilleren står i heisen |
| Spilleren er oppe hos bossen | Spilleren har dyttet vekk bossen, bossen har falt. Du har vunnet spillet! |