Oplossingen Oefeningen – Hoofdstuk 7: Lists

Oefening 7.10

```
from random import randint

LEEG = "."
SCHIP = "X"
AANTAL_SCHEPEN = 3
BREEDTE = 4
HOOGTE = 3


def toon_bord(spelbord):
    print(" ", end=" ")
    for kol in range(BREEDTE):
        print(chr(ord("A") + kol), end=" ")
    print()
    for rij in range(HOOGTE):
        print(rij + 1, end=" ")
        for kol in range(BREEDTE):
            print(spelbord[rij][kol], end=" ")
        print()


def is_geldige_plaats_voor_schip(spelbord, kol, rij):
    if spelbord[rij][kol] == SCHIP:
        return False
    if kol > 0 and spelbord[rij][kol - 1] == SCHIP:  # SCHIP LINKS
        return False
    if kol < len(spelbord[0]) - 1 and spelbord[rij][kol + 1] == SCHIP:  #
SCHIP RECHTS
        return False
    if rij > 0 and spelbord[rij - 1][kol] == SCHIP:  # SCHIP BOVEN
        return False
    if rij < len(spelbord) - 1 and spelbord[rij + 1][kol] == SCHIP:  #
SCHIP ONDER
        return False
    return True


def plaats_schepen(spelbord):
    te_plaatsen = AANTAL_SCHEPEN
    while te_plaatsen > 0:
        kol = randint(0, BREEDTE - 1)
        rij = randint(0, HOOGTE - 1)
        if is_geldige_plaats_voor_schip(spelbord, kol, rij):
            spelbord[rij][kol] = SCHIP
            te_plaatsen -= 1
```

```
def maak_spelbord(symbool):
    spelbord = []
    for i in range(HOOGTE):
        rij = BREEDTE * [symbool]
        spelbord.append(rij)
    return spelbord


spelbord = maak_spelbord(LEEG)
controle_bord = maak_spelbord(" ")
toon_bord(controle_bord)
plaats_schepen(spelbord)
#toon_bord(spelbord)
raak = 0
zetten = 0
while raak < AANTAL_SCHEPEN:
    cel = input("Kies een cel: ")
    rij = int(cel[1]) - 1
    kol = ord(cel[0]) - ord("A")
    if rij >= 0 and rij < HOOGTE and kol >= 0 and kol < BREEDTE:
        zetten += 1
        if spelbord[rij][kol] == SCHIP and controle_bord[rij][kol] == " ":
            raak += 1
            controle_bord[rij][kol] = SCHIP
            print("Raak!")
        else:
            controle_bord[rij][kol] = LEEG
    else:
        print("Ongeldig schot!")
    toon_bord(controle_bord)
print("Je gebruikte", zetten, "zetten")
toon_bord(spelbord)
```

ExtraOefening 7.1

```
invoer = input("Geef invoer:")
gegevens = invoer.split(" ")
klascode = gegevens[0]
voorschotbedrag = int(gegevens[1])
aantal_studenten = int(gegevens[2])
onkosten = aantal_studenten * [0]
student_nr = int(gegevens[3])
teller = 4
while student_nr != 0:
    onkosten[student_nr - 1] += int(gegevens[teller])
    student_nr = int(gegevens[teller + 1])
    teller += 2
print("Saldi voor klas", klascode)
for i in range(aantal_studenten):
    print("Student {}: onkosten: {} saldo: {}".format(i + 1, onkosten[i],
voorschotbedrag - onkosten[i]))
    print("Student {}: onkosten: {} saldo: {}".format(i + 1, onkosten[i],
voorschotbedrag - onkosten[i]))
```

ExtraOefening 7.3

```
GROOTTE = 7
matrix = []
for i in range(GROOTTE):
    matrix.append(GROOTTE * [0])

for i in range(GROOTTE):
    print("Geef de waarden voor kolom", i + 1)
    for j in range(GROOTTE):
        matrix[j][i] = int(input("Geef een waarde voor rij " + str(j + 1)))

for i in range(GROOTTE):
    for j in range(GROOTTE):
        print(matrix[i][j], end = " ")
    print()


som = 0
for i in range(GROOTTE):
    som += matrix[i][i]

print("Spoor: ", som)
```

ExtraOefening 7.4

```
def bereken_categorie(aantal_uren_gewerkt):
    if aantal_uren_gewerkt < 38:
        return 0
    if aantal_uren_gewerkt == 38:
        return 1
    overuren = aantal_uren_gewerkt - 38
    if overuren <= 10:
        return 2
    if overuren <= 15:
        return 3
    return 4


def bereken_loon(uurloon, aantal_uren_gewerkt):
    basisloon = 38 * uurloon
    if aantal_uren_gewerkt <= 38:
        return basisloon
    overuren = aantal_uren_gewerkt - 38
    if overuren > 15:
        basisloon += (overuren - 15) * 1.2 * uurloon
        overuren = 15
    if overuren > 10:
        basisloon += (overuren - 10) * 2 * uurloon
        overuren = 10
    if overuren > 0:
        basisloon += overuren * 1.5 * uurloon
    return basisloon


def bereken_terug_te_nemen_overuren(aantal_uren_gewerkt):
    if aantal_uren_gewerkt > 38 + 15:
        return aantal_uren_gewerkt - 38 - 15
    return 0


categorieen = [0] * 5
resultaten = []
totaal_loon = 0
totaal_terug_te_nemen_overuren = 0
personeelsnummer = input("Geef personeelsnummer: ")
while personeelsnummer != "0":
    uurloon = int(input("Geef uurloon: "))
    aantal_uren_gewerkt = int(input("Geef aantal uren gewerkt: "))
    categorieen[bereken_categorie(aantal_uren_gewerkt)] += 1
    loon = bereken_loon(uurloon, aantal_uren_gewerkt)
    totaal_loon += loon
    overuren = bereken_terug_te_nemen_overuren(aantal_uren_gewerkt)
    totaal_terug_te_nemen_overuren += overuren
    resultaten.append("{} {} {}".format(personeelsnummer, loon, overuren))
    personeelsnummer = input("Geef personeelsnummer: ")

for i in range(5):
    print("Categorie", i, ": ", categorieen[i])

print("Totaal overuren: ", totaal_terug_te_nemen_overuren)
print("Totaal loon: ", totaal_loon)
for resultaat in resultaten:
    print(resultaat)
```

ExtraOefening 7.5

```
def gewogen_gemiddelde(resultaten_vraag, aantal_personen):
    resultaat = 0
    for i in range(len(resultaten_vraag)):
        resultaat += (i + 1) * resultaten_vraag[i]
    return resultaat / aantal_personen


aantal_personen = 200
resultaten = [
    [25, 36, 55, 46, 38],
    [35, 45, 54, 65, 1],
    [75, 66, 44, 10, 5],
    [23, 33, 45, 50, 49]
]

hoogste_gewogen_gemiddelde = 0
vraag_hoogste_gewogen_gemiddelde = 0
for vraag in range(len(resultaten)):
    vraag_gewogen_gemiddelde = gewogen_gemiddelde(resultaten[vraag],
aantal_personen)
    print("Gewogen gemiddelde vraag {}: {:.2f}".format(vraag + 1,
vraag_gewogen_gemiddelde))
    if vraag_gewogen_gemiddelde > hoogste_gewogen_gemiddelde:
        vraag_hoogste_gewogen_gemiddelde = vraag + 1
        hoogste_gewogen_gemiddelde = vraag_gewogen_gemiddelde
print("Hoogste gewogen gemiddelde voor vraag
{}".format(vraag_hoogste_gewogen_gemiddelde))
```

ExtraOefening 7.8

```
def winnaar(bord, symbool):
    for i in range(3):
        if bord[i][0] == symbool and bord[i][1] == symbool and bord[i][2]
== symbool:
            return True
        if bord[0][i] == symbool and bord[1][i] == symbool and bord[2][i]
== symbool:
            return True
    if bord[0][0] == symbool and bord[1][1] == symbool and bord[2][2] ==
symbool:
        return True
    if bord[0][2] == symbool and bord[1][1] == symbool and bord[2][0] ==
symbool:
        return True
    return False


def einde_spel(bord):
    for i in range(3):
        for j in range(3):
            if bord[i][j] != "-":
                return False
    return True


def volgende(speler):
    speler = (speler + 1) % 2
    return speler


def toon_bord(spelbord):
    print("  1 2 3")
    for rij in range(3):
        print(rij + 1, end=" ")
        for kol in range(3):
            print(spelbord[rij][kol], end=" ")
        print()


def kies_rij_kolom():
    waarde = int(input("Geef een waarde van 1 tot 3:"))
    while waarde < 1 or waarde > 3:
        waarde = int(input("Ongeldige waarde. Geef een waarde van 1 tot
3:"))
    return waarde


def is_vrij(rij, kolom, bord):
    return bord[rij - 1][kolom - 1] == "-"


bord = [["-", "-", "-"], ["-", "-", "-"], ["-", "-", "-"]]
speler = 0
symbolen = "XO"
einde = False
while not einde:
    print("Beurt voor speler", speler + 1)
    rij = kies_rij_kolom()
    kolom = kies_rij_kolom()
```

```
while not is_vrij(rij, kolom, bord):
    print("Deze plaats is niet meer vrij.")
    rij = kies_rij_kolom()
    kolom = kies_rij_kolom()
print(symbolen[speler])
bord[rij - 1][kolom - 1] = symbolen[speler]
toon_bord(bord)
if winnaar(bord, symbolen[speler]):
    einde = True
    print("Speler", speler + 1, " heeft gewonnen")
elif einde_spel(bord):
    einde = True
    print("Gelijkspel")
speler = volgende(speler)
```