Зертханалық жұмыс №5
Функцияларды пайдаланып деректерді өңдеу бағдарламаларын әзірлеу

1. Жұмыстың мақсаты

Python тілінде функциялар құруды, оларға параметрлер беруді, return операторы арқылы есептеу нәтижесін қайтаруды және функцияларды көлік жүйелерінің өлшеу деректерін өңдеу үшін қолдануды үйрену.

2. Оқыту нәтижелері

Зертханалық жұмысты орындағаннан кейін білім алушы:

- def көмегімен функция құра алады;
- функцияны шақыра алады;
- параметр мен аргументтің айырмашылығын түсіндіре алады;
- return көмегімен нәтижені қайтара алады;
- бір функцияны бірнеше рет қолдана алады;
- тізімдерді функцияға аргумент ретінде бере алады;
- функциялар арқылы өлшеулерді талдай алады;
- бірнеше функциядан тұратын бағдарламаны құра алады.

3. Қысқаша теория

3.1 Функция дегеніміз не?

Функция (function) — белгілі бір тапсырманы орындайтын және қажет болған жағдайда нәтижені қайтаратын бағдарламаның жеке бөлігі.

Мысалы, автомобильдің жылдамдығын тексеретін бағдарламада бірдей шартты бірнеше рет жазудың орнына оны функцияға орналастыруға болады.

Функцияның жалпы құрылымы:

```
def function_name(parameter):
    # Функцияның командалары
    return result
```

Мұнда:

- def — функцияны анықтайтын кілт сөз;
- function_name — функция атауы;
- parameter — функция қабылдайтын параметр;
- return — нәтижені қайтаратын оператор.

3.2 Қарапайым функция

def hello():
print("Hello, Python!")hello()
Нәтиже:
Hello, Python!

Мұнда hello() функциясы анықталады, ал соңғы жолда шақырылады.

3.3 Параметр және аргумент
Мысалы:

```
def check_speed(speed):
    if speed > 90:
        print("HIGH")
    else:
        print("NORMAL")

check_speed(110)
check_speed(70)
```

Нәтиже:
HIGH
NORMAL

Бұл жерде speed — параметр, ал 110 және 70 — функцияға берілген аргументтер.
Бір функция әртүрлі мәндермен жұмыс істей алады.

3.4 return операторы

print() нәтижені экранға шығарады, ал return нәтижені функцияны шақырған жерге қайтарады.

```python
def calculate_average(a, b):
    average = (a + b) / 2
    return average

result = calculate_average(80, 100)

print("Average:", result)
```

Нәтиже:
Average: 90.0
return қолданудың артықшылығы — қайтарылған нәтижені басқа есептеулерде пайдалануға болады.

4. Тапсырма №1. Жылдамдықты тексеретін функция
Міндет: автомобиль жылдамдығын тексеретін check_speed() функциясын құру.
Шарттар:

Жылдамдық	**Күйі**
< 50 км/сағ	LOW
50–90 км/сағ	NORMAL
> 90 км/сағ	HIGH

Бағдарлама:

```
def check_speed(speed):

    if speed < 50:
        return "LOW"

    elif speed <= 90:
        return "NORMAL"

    else:
        return "HIGH"


print(check_speed(35))
print(check_speed(75))
print(check_speed(110))
```

Білім алушыға тапсырма: функцияны қосымша 0, 50, 90, 91 мәндерімен тексеріңіз.

5. Тапсырма №2. Өлшеулердің орташа мәнін есептейтін функция
Бұрын орташа мәнді былай есептедік:

```
temperature = [88, 92, 95, 101, 106]

average = sum(temperature) / len(temperature)
```

Енді осы операцияны функцияға орналастырамыз.

```
def calculate_average(data):
    average = sum(data) / len(data)
    return average


temperature = [88, 92, 95, 101, 106]

result = calculate_average(temperature)

print("Average temperature:", result)
```

Нәтиже:
Average temperature: 96.4
Бұл функцияны басқа параметрлер үшін де қолдануға болады:

```
speed = [40, 60, 80, 100]

print(calculate_average(speed))
```

Тапсырма: бір функцияны температура, жылдамдық және қысым тізімдері үшін қолданыңыз.

6. Тапсырма №3. Қозғалтқыш температурасын классификациялау
check_temperature() функциясын құрыңыз.
Шарттар:

Температура	**Күйі**
< 90°C	LOW
90–100°C	NORMAL
> 100°C	OVERHEATING

```python
def check_temperature(temp):

    if temp < 90:
        return "LOW"

    elif temp <= 100:
        return "NORMAL"

    else:
        return "OVERHEATING"


temperature = [88, 92, 95, 101, 106, 99, 87, 110]

for value in temperature:
    status = check_temperature(value)
    print(value, "°C ->", status)
```

Тексер!
Бұл мысалда екі тақырып біріктіріледі:
Тізім → for циклі → функция → нәтиже

7. Тапсырма №4. Қауіпті мәндерді анықтайтын функция
Міндет: өлшеулер тізімінен қауіпті мәндерді таңдайтын функция құру.
Мысалы, температура 100°C-тан жоғары болса, ол қауіпті деп есептеледі.

```
def find_dangerous(data):

    dangerous = []

    for value in data:
        if value > 100:
            dangerous.append(value)

    return dangerous


temperature = [88, 92, 95, 101, 106, 99, 87, 110]

result = find_dangerous(temperature)

print("Dangerous values:", result)
```

Нәтиже:
Dangerous values: [101, 106, 110]
Қосымша тапсырма: функцияны өзгертіп, қауіпті мәннің шегін де параметр ретінде беріңіз.

Мысалы:

```
def find_dangerous(data, limit):

    dangerous = []

    for value in data:
        if value > limit:
            dangerous.append(value)

    return dangerous


temperature = [88, 92, 95, 101, 106, 99, 87, 110]

print(find_dangerous(temperature, 100))
print(find_dangerous(temperature, 95))
```

Бір функция әртүрлі шекті мәндермен жұмыс істей алады.

8. Тапсырма №5. Бірнеше функциядан тұратын бағдарлама
Міндет: қозғалтқыш температурасын талдау үшін үш функция құру:

1. calculate_average(data) — орташа мәнді есептеу;
2. check_temperature(temp) — температура күйін анықтау;
3. find_dangerous(data) — қауіпті мәндерді жинау.

Толық бағдарлама:

```
def calculate_average(data):
    return sum(data) / len(data)


def check_temperature(temp):

    if temp < 90:
        return "LOW"

    elif temp <= 100:
        return "NORMAL"

    else:
        return "OVERHEATING"


def find_dangerous(data):

    dangerous = []

    for value in data:
        if value > 100:
            dangerous.append(value)

    return dangerous
```

```
temperature = [88, 92, 95, 101, 106, 99, 87, 110, 103, 94]

print("Measurements:", temperature)
print("Minimum:", min(temperature))
print("Maximum:", max(temperature))
print("Average:", calculate_average(temperature))

print()

for value in temperature:
    print(value, "->", check_temperature(value))

print()

dangerous = find_dangerous(temperature)

print("Dangerous values:", dangerous)
print("Number of dangerous values:", len(dangerous))

if len(dangerous) > 3:
    print("ENGINE CHECK REQUIRED")
else:
    print("ENGINE STATUS ACCEPTABLE")
```

Осы бағдарламада студенттер функциялардың негізгі артықшылығын көреді: әр функция белгілі бір жеке тапсырманы орындайды, ал негізгі бағдарлама олардың нәтижелерін біріктіреді.

9. Жеке тапсырмалар №1–10

Әр білім алушы өз нұсқасына сәйкес кемінде үш функциядан тұратын бағдарлама әзірлейді.

№	Тапсырма	Шарттар	Функциялар
1	Қозғалтқыш температурасын талдау	<90 LOW, 90–100 NORMAL, >100 OVERHEATING	check_temperature(), calculate_average(), find_dangerous()

2	Автомобиль жылдамдығын бақылау	<50 LOW, 50–90 NORMAL, >90 HIGH	check_speed(), calculate_average(), find_high_speed()
3	Май қысымын бақылау	<2 LOW, 2–4 NORMAL, >4 HIGH	check_pressure(), calculate_average(), find_low_pressure()
4	EV батареясының кернеуін талдау	<250 LOW, 250–450 NORMAL, >450 HIGH	check_voltage(), calculate_average(), find_abnormal()
5	Батарея зарядын бақылау	<20 LOW, 20–80 NORMAL, >80 HIGH	check_soc(), calculate_average(), find_low_soc()
6	Шина қысымын талдау	<2.0 LOW, 2.0–2.5 NORMAL, >2.5 HIGH	check_tire_pressure(), calculate_average(), find_abnormal()
7	EV батареясының температурасын бақылау	<15 LOW, 15–45 NORMAL, >45 HIGH	check_battery_temp(), calculate_average(), find_dangerous()
8	Отын шығынын талдау	<6 LOW, 6–10 NORMAL, >10 HIGH	check_consumption(), calculate_average(), find_high_consumption()
9	Қозғалтқыш айналымын талдау	<1000 LOW, 1000–4000 NORMAL, >4000 HIGH	check_rpm(), calculate_average(), find_high_rpm()
10	Салқындатқыш сұйықтық температурасын бақылау	<80 LOW, 80–100 NORMAL, >100 OVERHEATING	check_coolant(), calculate_average(), find_dangerous()

Әр нұсқаға қойылатын талаптар

Білім алушы:

1. 10 өлшеуден тұратын тізім құрады.
2. Бір өлшеуді классификациялайтын функция әзірлейді.

3. Орташа мәнді есептейтін функция әзірлейді.
4. Қауіпті немесе қалыпты диапазоннан тыс мәндерді қайтаратын функция әзірлейді.
5. for циклінде классификациялау функциясын шақырады.
6. Әр өлшеудің мәнін және күйін шығарады.
7. Қауіпті мәндердің жеке тізімін шығарады.
8. Қауіпті мәндер санын есептейді.
9. Жүйенің жалпы күйін анықтайды.
10. Нәтижелерге қорытынды жасайды.

10. Күрделендірілген жеке тапсырмалар №11–20

Бұл нұсқаларда студенттер төрт функциядан тұратын бағдарлама құрады. Кемінде бір функция екі немесе одан көп параметр қабылдауы тиіс.

№	Тапсырма	Күрделендірілген шарт
11	EV жоғары вольтты батареясының кернеуін бақылау	10 өлшеу. <300 LOW, 300–420 NORMAL, >420 HIGH. check_voltage(), calculate_average(), find_abnormal(), battery_status() функцияларын құру. Ауытқулар саны 3-тен көп болса → BATTERY CHECK REQUIRED.
12	EV батареясының температурасын бақылау	10 өлшеу. <10 TOO COLD, 10–45 NORMAL, >45 OVERHEATING. Максималды температура >55 болса → CRITICAL BATTERY TEMPERATURE. Қауіпті мәндерді қайтаратын функция құру.
13	Автомобиль жылдамдығын бақылау	10 өлшеу. 0 STOP, 1–60 NORMAL, 61–90 HIGH, >90 OVERSPEED. Жылдамдықты классификациялау, орташа мәнді есептеу, асып кетулерді жинау және олардың саны ≥3 болса → UNSAFE DRIVING.
14	Май қысымының қауіпті төмендеуін анықтау	10 өлшеу. <1.5 CRITICAL, 1.5–<2 LOW, 2–4 NORMAL, >4 HIGH. Критикалық мәндер саны мен тізімін қайтаратын функция құру. Кемінде бір критикалық мән болса → ENGINE CHECK REQUIRED.
15	Шина қысымының тұрақтылығын талдау	10 өлшеу. Қалыпты диапазон 2.1–2.5 bar. Ауытқулар тізімін құру, орташа қысымды есептеу, ауытқулар саны >2 болса → TIRE PRESSURE CHECK.

16	Отын шығынының тиімділігін бағалау	10 өлшеу. <6 ECONOMICAL, 6–9 NORMAL, >9 HIGH CONSUMPTION. Орташа шығын >9 болса → CHECK FUEL CONSUMPTION. Жоғары шығындар тізімін құру.
17	Қозғалтқыштың артық жүктелуін анықтау	10 өлшеу. <800 LOW, 800–3000 NORMAL, 3000–4500 HIGH, >4500 CRITICAL RPM. Критикалық мәндер саны ≥2 болса → ENGINE OVERLOAD.
18	EV батареясының заряд деңгейін бақылау	10 SOC мәні. <10 CRITICAL, 10–20 LOW, >20–80 NORMAL, >80 HIGH. Минималды SOC <10 болса → CHARGE IMMEDIATELY.
19	Салқындату жүйесінің тиімділігін талдау	10 температура. <70 COLD, 70–100 NORMAL, >100–110 HIGH, >110 CRITICAL. Максималды температура >110 болса → STOP ENGINE.
20	Кедергіге дейінгі қашықтықты бақылау	10 өлшеу, cm. <20 DANGER, 20–50 WARNING, >50–150 SAFE, >150 CLEAR. Минималды қашықтық <20 болса → EMERGENCY BRAKING.