МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

по дисциплине
«Информатика и программирование»

Студент
гр. БИН-25-2 _____ С.Е. Казюта

Ассистент
преподавателя _____ М.В. Водяницкий

Владивосток 2025

Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

Задание 1.

Имеется список объектов Фонда с указанием уровня угрозы:

“‘python objects = [(”Containment Cell A 4), (”Archive Vault 1), (”Bio Lab Sector 3), (”Observation Wing 2)] “‘

Используя ‘sorted‘ и лямбда-выражение, отсортируйте объекты по возрастанию уровня угрозы

Задание 2.

Дан список сотрудников Фонда с количеством проведенных смен и стоимостью одной смены:

“‘python staff-shifts = [(”name”: ”Dr. Shaw ”shift-cost”: 120, ”shifts”: 15), (”name”: ”Agent Torres ”shift-cost”: 90, ”shifts”: 22), (”name”: ”Researcher Hall ”shift-cost”: 150, ”shifts”: 10)] “‘

Используя ‘map‘ и лямбда-выражение, создайте список общей стоимости работы каждого сотрудника

Затем найдите максимальную стоимость с помощью ‘max‘

Задание 3.

Дан список персонала с уровнем допуска:

“‘python personnel = [(”name”: ”Dr. Klein ”clearance”: 2), (”name”: ”Agent Brooks ”clearance” 4), (”name”: ”Technician Reed ”clearance”: 1)] “‘

Используя ‘map‘ и лямбда-выражение, создайте новый список, где каждому сотруднику добавляется категория допуска:

* ‘”Restricted,, - уровень 1 * ‘”Confidential,, - уровни 2–3 * ‘”Top Secret,, - уровень 4 и выше

Результат должен быть списком словарей

Задание 4.

Дан список зон Фонда с указанием времени активности (в часах):

“‘python zones = [(”zone”: ”Sector-12 ”active-from”: 8, ”active-to”: 18), (”zone”: ”Deep Storage ”active-from”: 0, ”active-to”: 24), (”zone”: ”Research Wing ”active-from”: 9, ”active-to”: 17)] “‘

Используя ‘filter‘ и лямбда-выражение, выберите зоны, которые полностью работают в дневной период (с 8 до 18 включительно)

Задание 5.

Фонд анализирует служебные отчеты. Некоторые отчеты содержат внешние ссылки, которые должны быть удалены перед архивированием

“‘python reports = [(”author”: ”Dr. Moss ”text”: ”Analysis completed. Reference: http://external-archive.net”), (”author”: ”Agent Lee ”text”: ”Incident resolved without escalation.”), (”author”: ”Dr. Patel ”text”: ”Supplementary data available at https://secure-research.org”), (”author”: ”Supervisor Kane ”text”: ”No anomalies detected during inspection.”), (”author”: ”Researcher Bloom ”text”: ”Extended observations uploaded to http://research-notes.lab”), (”author”: ”Agent Novak ”text”: ”Perimeter secured. No external interference observed.”), (”author”: ”Dr. Hargreeve ”text”: ”Full containment log stored at https://internal-db.scp”), (”author”: ”Technician Moore ”text”: ”Routine maintenance completed successfully.”), (”author”: ”Dr. Alvarez ”text”: ”Cross-reference materials: http://crosslink.foundation”), (”author”: ”Security Officer Tan ”text”: ”Shift completed without incidents.”), (”author”: ”Analyst Wright ”text”: ”Statistical model published at https://analysis-hub.org”), (”author”: ”Dr. Kowalski ”text”: ”Behavioral deviations documented internally.”), (”author”: ”Agent Fischer ”text”: ”Additional footage archived: http://video-storage.sec”), (”author”: ”Senior Researcher Hall ”text”: ”All test results verified and approved.”), (”author”: ”Operations Lead Grant ”text”: ”Emergency protocol draft shared via https://ops-share.scp”)] “‘

Используя ‘filter‘ и лямбда-выражение:

1. Отберите отчеты, содержащие ссылки (‘http‘ или ‘https‘) 2. Преобразуйте их так, чтобы вместо ссылки отображалось ‘[ДАННЫЕ УДАЛЕНЫ]‘

Задание 6.

Дан список SCP-объектов с указанием их класса содержания:

“‘python scp-objects = [(”scp”: ”SCP-096 ”class”: ”Euclid”), (”scp”: ”SCP-173 ”class”: ”Euclid”), (”scp”: ”SCP-055 ”class”: ”Keter”), (”scp”: ”SCP-999 ”class”: ”Safe”), (”scp”: ”SCP-3001 ”class”: ”Keter”)] “‘

Используя ‘filter‘ и лямбда-выражение, сформируйте список SCP-объектов, которые требуют усиленных мер содержания

> □ К объектам с усиленными мерами относятся все SCP, **класс которых не равен ‘”Safe,,**

Результат должен быть списком словарей исходного формата

Задание 7.

Дан список инцидентов с количеством задействованного персонала:

“‘python incidents = [(”id”: 101, ”staff”: 4), (”id”: 102, ”staff”: 12), (”id”: 103, ”staff”: 7), (”id”: 104, ”staff”: 20)] “‘

Используя ‘sorted‘ и лямбда-выражение:

1. Отсортируйте инциденты по количеству персонала 2. Оставьте только три наиболее ресурсоемких инцидента

Задание 8.

Дан список протоколов безопасности и их уровней критичности:

“‘python protocols = [(”Lockdown 5), (”Evacuation 4), (”Data Wipe 3), (”Routine Scan 1)] “‘

Используя ‘map‘ и лямбда-выражение, создайте новый список строк вида:

“‘text ”Protocol Lockdown - Criticality 5„“

Задание 9.

Имеется список смен охраны с указанием длительности (в часах):

“‘python shifts = [6, 12, 8, 24, 10, 4] “‘

Используя ‘filter‘ и лямбда-выражение, выберите только те смены, которые:

* длятся не менее 8 часов * не превышают 12 часов

Задание 10.

Дан список сотрудников с результатами психологической оценки (от 0 до 100):

“‘python evaluations = [(”name”: ”Agent Cole ”score”: 78), (”name”: ”Dr. Weiss ”score”: 92), (”name”: ”Technician Moore ”score”: 61), (”name”: ”Researcher Lin ”score”: 88)] “‘

Используя ‘max‘ и лямбда-выражение, определите сотрудника с наивысшей оценкой Результатом должно быть имя сотрудника и его балл

Содержание

1 Выполнение работы

1.1 Задание 1

Задача данной программы это отсортировать объекты по возрастанию уровня угрозы.

Алгоритм работы:

1) Программа сортирует объекты по возрастанию уровня угрозы

2) Программа выводит список объектов.

```
objects = [
    ("Containment Cell A", 4),
    ("Archive Vault", 1),
    ("Bio Lab Sector", 3),
    ("Observation Wing", 2)
    ]

print(sorted(objects, key=lambda x: x[1]))
```

Рисунок 1 – Листинг программы для задания 1

Ключевые элементы кода:

1) objects = [] - список сортируемых объектов.

2) print(sorted(objects, key=lambda x: x[1])) - программа выводит сортированный список объектов по возрастанию.

1.2 Задание 2

Задача данной программы это найти максимальную стоимость работы сотрудника.

Алгоритм работы:

1) Программа расчитывает общую стоимость каждого сотрудника.

2) Программа выводит самую большую.

```
staff_shifts = [
    {"name": "Dr. Shaw", "shift_cost": 120, "shifts": 15},
    {"name": "Agent Torres", "shift_cost": 90, "shifts":
   22},
    {"name": "Researcher Hall", "shift_cost": 150, "shifts":
    10}
]

output = list(map(lambda a : a['shift_cost'] * a['shifts'],
   staff_shifts))

print(max(output))
```

Рисунок 2 – Листинг программы для задания 2

Ключевые элементы кода:

1) staff-shifts = [] - список сотрудников, их зарплат и смен.

2) output = list(map(lambda a : a[’shift-cost’] * a[’shifts’], staff-shifts)) - программа создаёт список зарплат умноженных на количество смен сотрудников.

3) print(max(output)) - программа выводит самое большое число.

1.3 Задание 3

Задача данной программы это создать новый списокна основе старого, но где каждому сотруднику добавляется категория допуска.

Алгоритм работы:

1) Программа создаёт новый список из сторого с нужными нам изменениями.

2) Программа выводит содержимое нового списка.

```
personnel = [
    {"name": "Dr. Klein", "clearance": 2},
    {"name": "Agent Brooks", "clearance": 4},
    {"name": "Technician Reed", "clearance": 1}
]
access_levels = {
    '1' : "Restricted" ,
    '2' : "Confidential",
    '3' : "Confidential",
    '4' : "Top Secret",
    '5' : "Top Secret",
    '6' : "Top Secret",
}

clearence_personnel = map(lambda a : {'name' : a['name'], "
   clearance" : access_levels[f"{a["clearance"]}"]},
   personnel)
for i in clearence_personnel:
    print(i)
```

Рисунок 3 – Листинг программы для задания 3

Ключевые элементы кода:

1) personnel = [] - список сотрудников и их категорий допуста (в числовом виде).

2) access-levels = () - словарь категорий доступа.

3) clearence-personnel = map(lambda a : (’name’ : a[’name’], ”clearance”: access-levels[f”(a[”clearance”])”]), personnel) - программа создаёт новый список из словарей с категориями доступа переведёнными из чисел в слова.

4) for i in clearence-personnel: print(i) - выводит значения списка

1.4 Задание 4

Задача данной программы это найти зоны которые полностью работают в дневной период (с 8 до 18 включительно).

Алгоритм работы:

1) Программа находит нужные нам зоны.

2) Программа выводит нужные нам зоны.

```
zones = [
    {"zone": "Sector-12", "active_from": 8, "active_to": 18},
    {"zone": "Deep Storage", "active_from": 0, "active_to": 24},
    {"zone": "Research Wing", "active_from": 9, "active_to": 17},
    ]

day_zones = list(filter(lambda x: x["active_from"] == 8 and x["active_to"] == 18, zones))
print(day_zones)
```

Рисунок 4 – Листинг программы для задания 4

Ключевые элементы кода:

1) zones = [] - список зон.

2) day-zones = list(filter(lambda x: x[”active-from”] == 8 and x[”active-to”] == 18, zones)) - программа находит все временные зоны работающие с 8 до 18.

3) print(day-zones) - программа выводит найденные ранее временные зоны.

1.5 Задание 5

Задача данной программы это заменить все ссылки в отчётах на ‘[ДАННЫЕ УДАЛЕНЫ]‘.

Алгоритм работы:

1) Программа находит все отчёты с ссылками.

2) Программа заменяет ссылки в отчёиах с ссылками на ‘[ДАННЫЕ УДАЛЕНЫ]‘.

3) Программа выводит список отредактированных отчётов.

```
reports = [
    {"author": "Dr. Moss", "text": "Analysis completed. Reference: http://external-archive.net"},
    {"author": "Agent Lee", "text": "Incident resolved without escalation."},
    {"author": "Dr. Patel", "text": "Supplementary data available at https://secure-research.org"},
    {"author": "Supervisor Kane", "text": "No anomalies detected during inspection."},
    {"author": "Researcher Bloom", "text": "Extended observations uploaded to http://research-notes.lab"},
    {"author": "Agent Novak", "text": "Perimeter secured. No external interference observed."},
    {"author": "Dr. Hargreeve", "text": "Full containment log stored at https://internal-db.scp"},
    {"author": "Technician Moore", "text": "Routine maintenance completed successfully."},
    {"author": "Dr. Alvarez", "text": "Cross-reference materials: http://crosslink.foundation"},
    {"author": "Security Officer Tan", "text": "Shift completed without incidents."},
    {"author": "Analyst Wright", "text": "Statistical model published at https://analysis-hub.org"},
    {"author": "Dr. Kowalski", "text": "Behavioral deviations documented internally."},
    {"author": "Agent Fischer", "text": "Additional footage archived: http://video-storage.sec"},
    {"author": "Senior Researcher Hall", "text": "All test results verified and approved."},
    {"author": "Operations Lead Grant", "text": "Emergency protocol draft shared via https://ops-share.scp"}
]

bad_reports = list(filter(lambda x: 'http' in x["text"], reports))
for i in bad_reports:
    i['text'] = f'{i['text'][:i['text'].index('http')]}{'ДАННЫЕ[ УДАЛЕНЫ]'}'
    print(i)
```

Рисунок 5 – Листинг программы для задания 5

Ключевые элементы кода:

1) reports = [] - список словарей с отчётами.

2) bad-reports = list(filter(lambda x: ’http’ in x[”text”], reports)) - программа находит все отчёты с ссылками.

3) for i in bad-reports: i[’text’] = f’(i[’text’][:i[’text’].index(’http’)])(’[ДАННЫЕ УДАЛЕНЫ]’)’ - программа заменяет отчёт на такой-же но с заменённой на ’[ДАННЫЕ УДАЛЕНЫ]’ ссылкой.

4) print(i) - программа выводит словарь с изменённым отчётом.

1.6 Задание 6

Задача данной программы это сформирование списка SCP-объектов, которые требуют усиленных мер содержания.

Алгоритм работы:

1) Программа фильтрует объекты по классам и выбирает всех кроме безобидных.

2) Программа формирует список объектов из оставшихся.

```
scp_objects = [
    {"scp": "SCP-096", "class": "Euclid"},
    {"scp": "SCP-173", "class": "Euclid"},
    {"scp": "SCP-055", "class": "Keter"},
    {"scp": "SCP-999", "class": "Safe"},
    {"scp": "SCP-3001", "class": "Keter"}
]

baddies = list(filter(lambda x: x["class"] != "Safe",
   scp_objects))
for i in baddies:
    print(i)
```

Рисунок 6 – Листинг программы для задания 6

Ключевые элементы кода:

1) scp-objects = [] - список объектов.

2) baddies = list(filter(lambda x: x[”class”] != ”Safe scp-objects)) - программа убирает из списка все объекты с классов ’Safe’.

3) for i in baddies: print(i) - программа выводит список оставшихся объектов.

1.7 Задание 7

Задача данной программы это нахождение трёх наиболее ресурсоемких инцидентов.

Алгоритм работы:

1) Программа сортирует инцидерны.

2) Программа выводит первые три из отсортированных.

```
incidents = [
    {"id": 101, "staff": 4},
    {"id": 102, "staff": 12},
    {"id": 103, "staff": 7},
    {"id": 104, "staff": 20}
]

new_incidents = sorted(incidents, key=lambda x: x['staff'],
   reverse=True)
for i in range(3):
    print(new_incidents[i])
```

Рисунок 7 – Листинг программы для задания 7

Ключевые элементы кода:

1) incidents = [] - список инцидентов.

2) new-incidents = sorted(incidents, key=lambda x: x[’staff’], reverse=True) - программа сортирует инциденты по количеству ушедших в мир иной в порядке уменьшения.

3) for i in range(3): print(new-incidents[i]) - программа выводит первые три объекта из списка.

1.8 Задание 8

Задача данной программы это создание нового списка f-строк на основе старого списка.

Алгоритм работы:

1) Программа создаёт список f-строк используя информацию из старого списка.

2) Программа выводит новый список.

```
protocols = [
    ("Lockdown", 5),
    ("Evacuation", 4),
    ("Data Wipe", 3),
    ("Routine Scan", 1)
]

output = list(map(lambda x : f'Protocol {x[0]} - Criticality
    {x[1]}', protocols))
for i in output:
    print(i)
```

Рисунок 8 – Листинг программы для задания 8

Ключевые элементы кода:

1) protocols = [] - список протоколов.

2) output = list(map(lambda x : f’Protocol (x[0]) - Criticality (x[1])’, protocols)) - программа создаёт список f-строк используя информацию из старого списка.

3) for i in output: print(i) - программа выводит новый список.

1.9 Задание 9

Задача данной программы это найти только те смены, которые: длятся не менее 8 часов, не превышают 12 часов.

Алгоритм работы:

1) Программа отфильтровывает среди всех смен только те, которые длятся не менее 8 часов и не превышают 12 часов.

2) Программа выводит отфильтрованный список смен.

```
shifts = [6, 12, 8, 24, 10, 4]

some_shifts = list(filter(lambda x : x >= 8 and x <= 12,
   shifts))
for i in some_shifts:
    print(i)
```

Рисунок 9 – Листинг программы для задания 9

Ключевые элементы кода:

1) shifts = [] - список длительности смен.

2) some-shifts = list(filter(lambda x : x >= 8 and x <= 12, shifts)) - программа фильтрует список длительностей смен из которых выберает только те, которые длятся не менее 8 часов и не превышают 12 часов.

3) for i in some-shifts: print(i) - пргрмма выводит список смен подходящих по нашим критериям.

1.10 Задание 10

Задача данной программы это имя и балл сотрудника с наивысшим баллом.

Алгоритм работы:

1) Программа находит сотрудника с нужной нам отличительной чертой.

2) Программа выводит данные сотрудника.

```
evaluations = [
    {"name": "Agent Cole", "score": 78},
    {"name": "Dr. Weiss", "score": 92},
    {"name": "Technician Moore", "score": 61},
    {"name": "Researcher Lin", "score": 88}
]
output = max(evaluations, key=lambda x : x['score'])
print(f'{output['name']}: {output['score']}')
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

1) evaluations = [] -список словарей синформацией об сотрудниках.

2) output = max(evaluations, key=lambda x : x[’score’]) - программа находит сотрудника с наивысшим баллом.

3) print(f’(output[’name’]): (output[’score’])’) - программа выводит f-строкой имя и балл сотрудника с наивысшим баллом.