

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 _____ К.Ф. Кучерчук
Ассистент
преподавателя _____ М.В. Водяницкий

Задание

Выполнить задания и оформить отчет по стандартам ВВГУ.

Основные задания

Задание 1

Имеется список объектов Фонда с указанием уровня угрозы:

```
1 objects = [
2     ("Containment Cell A", 4),
3     ("Archive Vault", 1),
4     ("Bio Lab Sector", 3),
5     ("Observation Wing", 2)
6 ]
7
```

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:

```
1 staff_shifts = [
2     {"name": "Dr. Shaw", "shift_cost": 120, "shifts": 15},
3     {"name": "Agent Torres", "shift_cost": 90, "shifts": 22},
4     {"name": "Researcher Hall", "shift_cost": 150, "shifts": 10}
5 ]
6
```

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:

```
1 personnel = [
2     {"name": "Dr. Klein", "clearance": 2},
3     {"name": "Agent Brooks", "clearance": 4},
4     {"name": "Technician Reed", "clearance": 1}
5 ]
```

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):

```
1 zones = [
2     {"zone": "Sector-12", "active_from": 8, "active_to": 18},
3     {"zone": "Deep Storage", "active_from": 0, "active_to": 24},
4     {"zone": "Research Wing", "active_from": 9, "active_to": 17}
5 ]
6
```

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.

Приведён список из 15 отчётов (см. исходный файл). Используя `filter` и лямбда-выражение:

1. Отберите отчёты, 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:

```
1 scp_objects = [
2     {"scp": "SCP-096", "class": "Euclid"},
3     {"scp": "SCP-173", "class": "Euclid"},
4     {"scp": "SCP-055", "class": "Keter"},
```

```
5 {"scp": "SCP-999", "class": "Safe"},  
6 {"scp": "SCP-3001", "class": "Keter"}  
7 ]  
8
```

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:

```
1 incidents = [  
2     {"id": 101, "staff": 4},  
3     {"id": 102, "staff": 12},  
4     {"id": 103, "staff": 7},  
5     {"id": 104, "staff": 20}  
6 ]  
7
```

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:

```
1 protocols = [  
2     ("Lockdown", 5),  
3     ("Evacuation", 4),  
4     ("Data Wipe", 3),  
5     ("Routine Scan", 1)  
6 ]  
7
```

Используя `map` и лямбда-выражение, создайте новый список строк вида: "Protocol Lockdown - Criticality 5"

Задание 9

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

```
1 shifts = [6, 12, 8, 24, 10, 4]  
2
```

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

- делятся не менее 8 часов
- не превышают 12 часов

Задание 10

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

```
1 evaluations = [  
2     {"name": "Agent Cole", "score": 78},  
3     {"name": "Dr. Weiss", "score": 92},  
4     {"name": "Technician Moore", "score": 61},  
5     {"name": "Researcher Lin", "score": 88}  
6 ]  
7
```

Используя `max` и лямбда-выражение, определите сотрудника с наивысшей оценкой.

Результатом должно быть имя сотрудника и его балл.

Содержание

1 Выполнение работы	3
1.1 Задание 1	3
1.2 Задание 2	3
1.3 Задание 3	3
1.4 Задание 4	4
1.5 Задание 5	4
1.6 Задание 6	7
1.7 Задание 7	7
1.8 Задание 8	7
1.9 Задание 9	8
1.10 Задание 10	8

1 Выполнение работы

1.1 Задание 1

Требуется отсортировать список кортежей, представляющих объекты Фонда, по возрастанию уровня угрозы (второй элемент кортежа). Для этого используется встроенная функция `sorted` с ключом сортировки, заданным лямбда-выражением.

```

1 objects = [
2     ("Containment Cell A", 4),
3     ("Archive Vault", 1),
4     ("Bio Lab Sector", 3),
5     ("Observation Wing", 2)
6 ]
7
8 objects = [sorted(objects, key=lambda x : x[1], reverse=
    False)]

```

Рисунок 1 – Листинг программы для задания 1

Функция `sorted` принимает список `objects` и ключ сортировки `key=lambda x: x[1]`, который извлекает второй элемент каждого кортежа. Параметр `reverse=False` (по умолчанию) обеспечивает сортировку по возрастанию. Результат — новый список объектов, упорядоченный от наименее угрожающего к наиболее угрожающему.

1.2 Задание 2

Необходимо рассчитать общую стоимость работы каждого сотрудника как произведение количества смен на стоимость одной смены, а затем найти максимальное значение среди всех сотрудников.

```

1 staff_shifts = [
2     {"name": "Dr. Shaw", "shift_cost": 120, "shifts": 15},
3     {"name": "Agent Torres", "shift_cost": 90, "shifts": 22},
4     {"name": "Researcher Hall", "shift_cost": 150, "shifts": 10}
5 ]
6
7 total_shifts = [*map(lambda x: x['shift_cost'] * x['shifts'],
8                     staff_shifts)]
9 result = max(total_shifts)

```

Рисунок 2 – Листинг программы для задания 2

С помощью `map` и лямбда-функции `lambda x: x['shift_cost'] * x['shifts']` вычисляется общая стоимость для каждого сотрудника. Результат преобразуется в список `total_shifts`. Затем функция `max` находит наибольшее значение в этом списке. Отметим, что в текущей реализации возвращается только максимальная сумма, без указания, кому она принадлежит.

1.3 Задание 3

Требуется преобразовать числовые значения уровня допуска в строковые категории: «Restricted», «Confidential» или «Top Secret», согласно заданному отображению.

```

1 personnel = [
2     {"name": "Dr. Klein", "clearance": 2},
3     {"name": "Agent Brooks", "clearance": 4},
4     {"name": "Technician Reed", "clearance": 1}
5 ]
6
7 clearances = {
8     1 : "Restricted",
9     2 : "Confidential",
10    3 : "Confidential",
11    4 : "Top Secret"
12 }
13
14 lst = [*map(lambda x: clearances.get(x['clearance']),
15             personnel)]
16 for i in range(len(lst)):
17     personnel[i]['clearance'] = lst[i]

```

Рисунок 3 – Листинг программы для задания 3

Сначала создаётся словарь `clearances`, задающий соответствие уровня допуска его названию. Затем с помощью `map` и лямбда-выражения `lambda x: clearances.get(x['clearance'])` формируется список строковых категорий. Далее исходный список словарей обновляется: числовое значение `clearance` заменяется на строковое. Такой подход изменяет исходные данные на месте.

1.4 Задание 4

Необходимо отфильтровать список зон, оставив только те, которые активны в течение всего дневного периода — с 8:00 до 18:00 включительно. Это означает, что зона должна начинать работу не позже 8:00 и заканчивать не раньше 18:00.

```

1 zones = [
2     {"zone": "Sector-12", "active_from": 8, "active_to": 18},
3     {"zone": "Deep Storage", "active_from": 0, "active_to": 24},
4     {"zone": "Research Wing", "active_from": 9, "active_to": 17}
5 ]
6 zones = [*filter(lambda x: x['active_from'] <= 8 and x['active_to'] >= 18, zones)]

```

Рисунок 4 – Листинг программы для задания 4

Функция `filter` применяет лямбда-условие `x['active_from'] <= 8 and x['active_to'] >= 18`, проверяющее полное покрытие дневного интервала. Например, зона «Sector-12» (8–18) проходит фильтр, а «Research Wing» (9–17) — нет. Результат преобразуется в список, содержащий только подходящие зоны.

1.5 Задание 5

Требуется извлечь отчёты, содержащие URL-ссылки, и заменить каждую ссылку на строку [ДАННЫЕ УДАЛЕНЫ], сохранив остальной текст без изменений.

```

1 reports = [
2     {"author": "Dr. Moss", "text": "Analysis completed.  
Reference: http://external-archive.net"},  

3     {"author": "Agent Lee", "text": "Incident resolved  
without escalation."},  

4     {"author": "Dr. Patel", "text": "Supplementary data  
available at https://secure-research.org"},  

5     {"author": "Supervisor Kane", "text": "No anomalies  
detected during inspection."},  

6     {"author": "Researcher Bloom", "text": "Extended  
observations uploaded to http://research-notes.lab"},  

7     {"author": "Agent Novak", "text": "Perimeter secured. No  
external interference observed."},  

8     {"author": "Dr. Hargreeve", "text": "Full containment  
log stored at https://internal-db.scp"},  

9     {"author": "Technician Moore", "text": "Routine  
maintenance completed successfully."},  

10    {"author": "Dr. Alvarez", "text": "Cross-reference  
materials: http://crosslink.foundation"},  

11    {"author": "Security Officer Tan", "text": "Shift  
completed without incidents."},  

12    {"author": "Analyst Wright", "text": "Statistical model  
published at https://analysis-hub.org"},  

13    {"author": "Dr. Kowalski", "text": "Behavioral  
deviations documented internally."},  

14    {"author": "Agent Fischer", "text": "Additional footage  
archived: http://video-storage.sec"},  

15    {"author": "Senior Researcher Hall", "text": "All test  
results verified and approved."},  

16    {"author": "Operations Lead Grant", "text": "Emergency  
protocol draft shared via https://ops-share.scp"}  

17 ]
18
19 # фильтр отчётов
20 reports_with_links = list(filter(
21     lambda r: "http://" in r["text"] or "https://" in r["  
text"],  

22     reports
23 ))
24
25 # замена ссылок
26 def remove_urls_from_text(text):
27     words = text.split()
28     new_words = []
29     for word in words:
30         if word.startswith("http://") or word.startswith("https://"):
31             new_words.append("ДАННЫЕ[ УДАЛЕНЫ]")
32         else:
33             new_words.append(word)
34     return " ".join(new_words)
35
36 # преобразование отчётов
37 cleaned_reports = list(map(
38     lambda r: {
39         "author": r["author"],
40         "text": remove_urls_from_text(r["text"])
41     },
42     reports_with_links
43 )))

```

Рисунок 5 – Листинг программы для задания 5

Сначала `filter` отбирает отчёты, в тексте которых присутствует `http://` или `https://`. Затем определяется вспомогательная функция `remove_urls_from_text`, ко-

торая разбивает текст на слова и заменяет слова, начинающиеся с `http`, на заглушку. Наконец, `map` применяет эту функцию ко всем отфильтрованным отчётом, формируя новый список с очищенным текстом.

1.6 Задание 6

Следует отфильтровать SCP-объекты, оставив только те, чей класс не равен `"Safe"`, так как именно они требуют усиленных мер содержания.

```

1 scp_objects = [
2     {"scp": "SCP-096", "class": "Euclid"}, 
3     {"scp": "SCP-173", "class": "Euclid"}, 
4     {"scp": "SCP-055", "class": "Keter"}, 
5     {"scp": "SCP-999", "class": "Safe"}, 
6     {"scp": "SCP-3001", "class": "Keter"} 
7 ]
8
9 res_list = [*filter(lambda x: x['class'] != 'Safe',
10                     scp_objects)]

```

Рисунок 6 – Листинг программы для задания 6

Функция `filter` с лямбда-условием `x['class'] != 'Safe'` отбирает объекты классов «`Euclid`» и «`Keter`». Результат преобразуется в список, содержащий исходные словари без изменений, что соответствует требованию.

1.7 Задание 7

Необходимо отсортировать инциденты по количеству задействованного персонала в порядке убывания и выбрать три первых (наиболее ресурсоёмких).

```

1 incidents = [
2     {"id": 101, "staff": 4}, 
3     {"id": 102, "staff": 12}, 
4     {"id": 103, "staff": 7}, 
5     {"id": 104, "staff": 20} 
6 ]
7
8 sorted_incidents = sorted(incidents, key=lambda x: x['staff'],
9                             reverse=True)
10 incidents = sorted_incidents[:3]

```

Рисунок 7 – Листинг программы для задания 7

Сначала список `incidents` сортируется по ключу `x['staff']` с параметром `reverse=True`, чтобы самые крупные инциденты оказались в начале. Затем берётся срез первых трёх элементов с помощью `[:3]`. Результат — список из трёх словарей с наибольшим значением `staff`.

1.8 Задание 8

Требуется преобразовать список кортежей (протокол, критичность) в список строк заданного формата.

```

1 protocols = [
2     ("Lockdown", 5),
3     ("Evacuation", 4),
4     ("Data Wipe", 3),
5     ("Routine Scan", 1)
6 ]
7
8 protocols = [*map(lambda obj: f'Protocol {obj[0]} - 
Criticality {obj[1]}', protocols)]

```

Рисунок 8 – Листинг программы для задания 8

Функция `map` применяет лямбда-выражение `lambda obj: f'Protocol {obj[0]} - Criticality {obj[1]}'`, которое формирует строку для каждого кортежа. Результат преобразуется в список строк, соответствующих требуемому шаблону.

1.9 Задание 9

Необходимо отфильтровать список длительностей смен, оставив только те, что находятся в диапазоне от 8 до 12 часов включительно.

```

1 shifts = [6, 12, 8, 24, 10, 4]
2
3 shifts = [*filter(lambda x: 8 <= x <= 12, shifts)]

```

Рисунок 9 – Листинг программы для задания 9

Функция `filter` с условием `8 <= x <= 12` отбирает подходящие значения. Например, из исходного списка `[6, 12, 8, 24, 10, 4]` будут выбраны `12, 8, 10`, так как они удовлетворяют условию. Результат — новый список, содержащий только допустимые длительности смен.

1.10 Задание 10

Требуется найти сотрудника с наивысшей оценкой и вывести его имя и балл.

```

1 evaluations = [
2     {"name": "Agent Cole", "score": 78},
3     {"name": "Dr. Weiss", "score": 92},
4     {"name": "Technician Moore", "score": 61},
5     {"name": "Researcher Lin", "score": 88}
6 ]
7
8 best_chel = max(evaluations, key=lambda x: x['score'])
9 print(f'Сотрудник {best_chel["name"]} имеет высший балл
равный {best_chel["score"]}')

```

Рисунок 10 – Листинг программы для задания 10

Функция `max` принимает список словарей `evaluations` и ключ `key=lambda x: x['score']`, по которому происходит сравнение. Возвращается словарь сотрудника с максимальным баллом. Затем с помощью `f`-строки формируется и выводится сообщение

с именем и оценкой. В данном случае выводится: «Сотрудник Dr. Weiss имеет высший балл равный 92».