

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 основы программирования»

Студент		
гр. БИН-25-3	_____	В.Е. Соловьева
Ассистент		
преподавателя	_____	М.В. Водяницкий

Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

Задание 1. Имеется список объектов Фонда с указанием уровня угрозы:

```
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:

```
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:

```
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:

- 1) "Restricted уровень 1
- 2) "Confidential уровни 2–3
- 3) "Top Secret уровень 4 и выше

Результат должен быть списком словарей

Задание 4. Дан список зон Фонда с указанием времени активности (в часах):

```
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

```
reports = [  
    {"author": "Dr. Moss", "text": "Analysis completed. Reference: http://external"},  
    {"author": "Agent Lee", "text": "Incident resolved without escalation."},  
    {"author": "Dr. Patel", "text": "Supplementary data available at https://secure"},  
    {"author": "Supervisor Kane", "text": "No anomalies detected during inspection"},  
    {"author": "Researcher Bloom", "text": "Extended observations uploaded to http"},  
    {"author": "Agent Novak", "text": "Perimeter secured. No external interference"},  
    {"author": "Dr. Hargreeve", "text": "Full containment log stored at https://int"},  
    {"author": "Technician Moore", "text": "Routine maintenance completed successf"},  
    {"author": "Dr. Alvarez", "text": "Cross-reference materials: http://crosslink"},  
    {"author": "Security Officer Tan", "text": "Shift completed without incidents."},  
    {"author": "Analyst Wright", "text": "Statistical model published at https://ar"},  
    {"author": "Dr. Kowalski", "text": "Behavioral deviations documented internall"},  
    {"author": "Agent Fischer", "text": "Additional footage archived: http://video"},  
    {"author": "Senior Researcher Hall", "text": "All test results verified and app
```

```
{"author": "Operations Lead Grant", "text": "Emergency protocol draft shared via  
"]
```

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:

```
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:

```
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:

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

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):

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

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

- 1) длятся не менее 8 часов
- 2) не превышают 12 часов

Задание 10. Дан список сотрудников с результатами психологической оценки (от 0 до 100):

```
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

Оформление отчета. Отчет оформляется строго по СТО - скачать требования можно с сайта ВВГУ, либо отсюда

Не забудьте добавить страницу "Задание" с копией содержимого этого файла (с правильным оформлением списков и т.д.)

В отчете должно быть объяснено как работает ваша программа (каждое отдельное задание)

Содержание

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

1 Выполнение работы

1.1 Задание 1

В данном задании необходимо, используя `sorted` и лямбда-выражение, отсортировать объекты по возрастанию уровня угрозы.

На рисунке 1 представлен код полученной программы.

```

1 objects = [
2     ("Containment Cell A", 4),
3     ("Archive Vault", 1),
4     ("Bio Lab Sector", 3),
5     ("Observation Wing", 2)
6 ]
7
8 sorted_objects = sorted(objects, key=lambda x: x[1])
9
10 print(sorted_objects)

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Существует список кортежей `objects`, где каждый кортеж содержит название объекта и его уровень угрозы;
- 2) Используется функция `sorted()` с параметром `key=lambda x: x[1]`, которая указывает, что сортировка должна выполняться по уровню угрозы;
- 3) Результат выводится в консоль.

Таким образом задание выполнено.

1.2 Задание 2

В данном задании необходимо, используя `map` и лямбда-выражение, создать список общей стоимости работы каждого сотрудника. И затем найти максимальную стоимость с помощью `max`.

На рисунке 2 представлен код программы.

```

1 staff_shifts = [
2     {"name": "Dr. Shaw", "shift_cost": 120, "shifts": 15},
3     {"name": "Agent Torres", "shift_cost": 90, "shifts":
4         22},
5     {"name": "Researcher Hall", "shift_cost": 150, "shifts":
6         10}
7 ]
8
9 total_costs = list(map(lambda x: {'name': x['name'], '
10     total_cost': x['shift_cost'] * x['shifts']}, staff_shifts
11 ))
12
13 max_cost = max(total_costs, key=lambda x: x['total_cost'])
14
15 print(total_costs)
16 print(max_cost)

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит сотрудника, количество проведенный смен и стоимость одной смены;
- 2) С помощью функции `map()` и лямбда-выражения для каждого сотрудника создается словарь с именем и общей стоимостью;
- 3) С помощью функции `max()` находится максимальная стоимость;
- 4) В консоль выводится общая стоимость работы каждого сотрудника и максимальная стоимость.

Таким образом задание выполнено.

1.3 Задание 3

В данном задании необходимо, используя `map` и лямбда-выражение, создать новый список, где каждому сотруднику добавляется категория допуска.

На рисунке 3 представлен код программы.

```

1 personnel = [
2     {"name": "Dr. Klein", "clearance": 2},
3     {"name": "Agent Brooks", "clearance": 4},
4     {"name": "Technician Reed", "clearance": 1}
5 ]
6
7 with_category = list(map(lambda x: {'name': x['name'], '
8     clearance': x['clearance'], 'category': (
9     'Restricted' if x['clearance'] == 1 else
10    'Confidential' if 2 <= x['clearance'] <= 3 else
11    'Top Secret'}), personnel))
12 print(with_category)

```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит сотрудника и его уровень допуска;
- 2) С помощью функции `map()` и лямбда-выражения добавляется категория допуска;
- 3) В консоль выводится новый список словарей.

Таким образом задание выполнено.

1.4 Задание 4

В данном задании необходимо, используя `filter` и лямбда-выражение, выбрать зоны, которые полностью работают в дневной период (с 8 до 18 включительно).

На рисунке 4 представлен код решения.

```
1 zones = [  
2     {"zone": "Sector-12", "active_from": 8, "active_to":  
3     18},  
4     {"zone": "Deep Storage", "active_from": 0, "active_to":  
5     24},  
6     {"zone": "Research Wing", "active_from": 9, "active_to":  
7     17}  
8 ]  
9 daytime_zones = list(filter(lambda x: x['active_from'] >= 8  
10    and x['active_to'] <= 18, zones))  
11 print(daytime_zones)
```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит зону Фонда и время активности (в часах);
- 2) С помощью функции `filter()` и лямбда-выражения создается список словарей зон, которые полностью работают в дневной период (с 8 до 18 включительно);
- 3) Новый список словарей выводится в консоль.

Таким образом задание выполнено.

1.5 Задание 5

В данном задании требуется, используя `filter` и лямбда-выражение, отобразить отчеты, содержащие ссылки и преобразовать их так, чтобы вместо ссылки отображалось [ДАННЫЕ УДАЛЕНЫ].

На рисунке 5 представлен код программы.

```

1 reports = [
2     {"author": "Dr. Moss", "text": "Analysis completed.
3     Reference: http://external-archive.net"},
4     {"author": "Agent Lee", "text": "Incident resolved
5     without escalation."},
6     {"author": "Dr. Patel", "text": "Supplementary data
7     available at https://secure-research.org"},
8     {"author": "Supervisor Kane", "text": "No anomalies
9     detected during inspection."},
10    {"author": "Researcher Bloom", "text": "Extended
11    observations uploaded to http://research-notes.lab"},
12    {"author": "Agent Novak", "text": "Perimeter secured. No
13    external interference observed."},
14    {"author": "Dr. Hargreeve", "text": "Full containment
15    log stored at https://internal-db.scp"},
16    {"author": "Technician Moore", "text": "Routine
17    maintenance completed successfully."},
18    {"author": "Dr. Alvarez", "text": "Cross-reference
19    materials: http://crosslink.foundation"},
20    {"author": "Security Officer Tan", "text": "Shift
21    completed without incidents."},
22    {"author": "Analyst Wright", "text": "Statistical model
23    published at https://analysis-hub.org"},
24    {"author": "Dr. Kowalski", "text": "Behavioral
25    deviations documented internally."},
26    {"author": "Agent Fischer", "text": "Additional footage
27    archived: http://video-storage.sec"},
28    {"author": "Senior Researcher Hall", "text": "All test
29    results verified and approved."},
30    {"author": "Operations Lead Grant", "text": "Emergency
31    protocol draft shared via https://ops-share.scp"}
32 ]
33
34 with_links = list(filter(lambda x: 'http://' in x['text'] or
35                          'https://' in x['text'], reports))
36
37 print(with_links)
38
39 import re
40
41 cleaned = list(map(lambda x: {'author': x['author'], 'text':
42                             re.sub(r'https?:\/\/[^\s]+', 'ДАННЫЕ[ УДАЛЕНЫ]', x['text'])
43                             }, with_links))
44
45 print(cleaned)

```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит автора и отчеты, некоторые из которых содержат ссылки;
- 2) С помощью функции `filter()` и лямбда-выражения создается список словарей, содержащий только отчеты с ссылками;
- 3) Список словарей, содержащий только отчеты с ссылками выводится в консоль;
- 4) Импортируется библиотека `re`;
- 5) С помощью функции `map()`, лямбда-выражения и библиотеки `re` ссылки создается новый список словарей, где ссылки заменены на [ДАННЫЕ УДАЛЕНЫ];
- 6) Список словарей, где ссылки заменены на [ДАННЫЕ УДАЛЕНЫ] выводится в консоль.

Таким образом задание выполнено.

1.6 Задание 6

В данном задании требуется, используя `filter` и лямбда-выражение, сформировать список SCP-объектов, которые требуют усиленных мер содержания.

На рисунке 6 представлен код программы.

```

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 enhanced = list(filter(lambda x: x['class'] != 'Safe',
10                        scp_objects))
11 print(enhanced)

```

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит SCP-объект и его класс содержания;
- 2) С помощью функции `filter()` и лямбда-выражения создается новый список словарей SCP-объектов, которые требуют усиленных мер содержания;
- 3) Новый список словарей выводится в консоль.

Таким образом задание выполнено.

1.7 Задание 7

В данном задании требуется, используя `sorted` и лямбда-выражение, отсортировать инциденты по количеству персонала и оставить только три наиболее ресурсоемких инцидента.

На рисунке 7 представлен код программы.

```
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  
11 print(sorted_incidents)  
12  
13 top3 = sorted_incidents[:3]  
14 print(top3)
```

Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит ID инцидента и количество задействованного персонала;
- 2) С помощью функции `sorted()` и лямбда-выражения создается новый список словарей, который является отсортированным изначальным списком;
- 3) Новый список словарей выводится в консоль;
- 4) С помощью среза выделяются три наиболее ресурсоемких инцидента;
- 5) Список словарей, содержащий три наиболее ресурсоемких инцидента выводится в консоль.

Таким образом задание выполнено.

1.8 Задание 8

В данном задании требуется, используя `map` и лямбда-выражение, создать новый список вида:

"Protocol Lockdown - Criticality 5"

На рисунке 8 представлен код программы.

```

1 protocols = [
2     ("Lockdown", 5),
3     ("Evacuation", 4),
4     ("Data Wipe", 3),
5     ("Routine Scan", 1)
6 ]
7
8 new = list(map(lambda x: f'Protocol {x[0]} - Criticality {x
9     [1]}', protocols))
10 print(new)

```

Рисунок 8 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Существует список кортежей, где каждый кортеж содержит протокол безопасности и его уровень критичности;
- 2) С помощью функции `map()` и лямбда-выражения создается новый список кортежей нужного вида;
- 3) Новый список кортежей выводится в консоль.

Таким образом задание выполнено.

1.9 Задание 9

В данном задании требуется, используя `filter` и лямбда-выражение, выбрать только те смены, которые:

- 1) длятся не менее 8 часов
- 2) не превышают 12 часов

На рисунке 9 представлен код программы.

```

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

```

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Существует список смен охраны с указанием длительности (в часах);
- 2) С помощью функции `filter()` и лямбда-выражения создается новый список, который содержит смены с длительностью от 8 до 12 часов;
- 3) Новый список выводится в консоль.

Таким образом задание выполнено.

1.10 Задание 10

В данном задании требуется, используя `max` и лямбда-выражение, определить сотрудника с наивысшей оценкой.

На рисунке 10 представлен код программы.

```
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 top = max(evaluations, key=lambda x: x['score'])  
9  
10 print(top)
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Существует список словарей, где каждый словарь содержит сотрудника и результаты его психологической оценки;
- 2) С помощью функции `max()` и лямбда-выражения создается новый список словарей, содержащий сотрудника с наивысшей оценкой и его балл;
- 3) Новый список словарей выводится в консоль.

Таким образом задание выполнено.