

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. БИС-25-3
Ассистент
преподавателя

Д. А. Пестерев
М. В. Водяницкий

Задание

Выполнить задания на Python и оформить отчет по стандартам ВВГУ.

Задание 1. Используя sorted и лямбда-выражение, отсортируйте объекты по возрастанию уровня угрозы.

Задание 2. Используя map и лямбда-выражение, создайте список общей стоимости работы каждого сотрудника. Затем найдите максимальную стоимость с помощью max.

Задание 3. 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. Используя filter и лямбда-выражение, выберите зоны, которые полностью работают в дневной период (с 8 до 18 включительно).

Задание 5. Используя filter и лямбда-выражение:

- Отберите отчеты, содержащие ссылки ([http](http://) или [https](https://))
- Преобразуйте их так, чтобы вместо ссылки отображалось [ДАННЫЕ УДАЛЕНЫ]

Задание 6. Используя filter и лямбда-выражение, сформируйте список SCP-объектов, которые требуют усиленных мер содержания

Задание 7. Используя sorted и лямбда-выражение:

- Отсортируйте инциденты по количеству персонала
- Оставьте только три наиболее ресурсоемких инцидента

Задание 8. Используя map и лямбда-выражение, создайте новый список строк вида: "Protocol Lockdown - Criticality 5"

Задание 9. Используя filter и лямбда-выражение, выберите только те смены, которые:

- делятся не менее 8 часов
- не превышают 12 часов

Задание 10. Используя max и лямбда-выражение, определите сотрудника с наивысшей оценкой. Результатом должно быть имя сотрудника и его балл.

Оглавление

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

1 Выполнение работы

1.1 Задание 1

В задании необходимо было написать программу, которая с помощью sorted и лямбда-выражения, отсортирует объекты списка по возрастанию уровня угрозы. На рисунке 1 представлен код программы.

```

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

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Дан список кортежей объектов Фонда с указанием уровня угрозы.
- 2) Далее через print выводим: функцию sorted, на вход которой подаём наш список, а также функция key, по результату которой идёт сравнение, мы передаём лямбда-выражение в котором обращаемся ко второму элементу кортежа.

1.2 Задание 2

В данном задании необходимо было написать программу, которая перебирает список и выводит на экран максимальную стоимость смен сотрудника. На рисунке 2 представлен код программы.

```

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

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

1) Дан список словарей сотрудников Фонда с количеством проведенных смен и стоимостью одной смены.

2) Через print выводим: максимальное значение функции map, которая применяет одну функцию к каждому элементу последовательности и возвращает результат, в неё мы передаём наше лямбда-выражение, для каждого x: количество смен умноженное на стоимость одной смены, а также передаём наш список.

1.3 Задание 3

В данном задании необходимо было написать программу которая добавляет в словарь каждого сотрудника категорию доступа в соответствии с уровнем. На рисунке 3 представлен код программы.

```

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

```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

1) Дан список персонала с уровнем допуска.

2) Через print выводим: список с помощью функции list в котором функция map которая применяет к каждому элементу лямбда-выражение: для каждого x: распаковываем словарь при создании нового словаря и добавляем новый ключ со значением соответствующим уровню допуска.

1.4 Задание 4

В этом задании необходимо было написать программу, которая фильтрует зоны по временным значениям. На рисунке 4 представлен код программы.

```

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

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Дан список зон Фонда с указанием времени активности.
- 2) Через print выводим: список с помощью функции list в котором функция filter с помощью лямбда-выражения фильтрует словари по времеени, для каждого элемента от 8 часов до 18 включительно.

1.5 Задание 5

В данном задании необходимо было написать программу, которая будет проверять пароль, введённый пользователем с консоли, на надёжность. На рисунке 5 представлен код программы.

```

1 # Задание 5
2 import re
3
4 reports = [
5     {"author": "Dr. Moss", "text": "Analysis completed. Reference: http://external-archive.net"},  

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

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

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

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

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

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

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

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

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

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

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

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

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

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

20 ]
21 ls = list(filter(lambda x: 'http://' in x['text'] or 'https://' in x['text'], reports))
22 for report in ls:
23     report['text'] = re.sub( pattern: 'https?://\S+', repl: '[ДАННЫЕ УДАЛЕНЫ]', report['text'])
24     print(report)

```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Дан список служебных отчётов, некоторые из которых содержат ссылки.
- 2) Создаём список в который с помощью функции filter отбираем отчёты содержащие ссылки.
- 3) Далее перебираем отчёты в созданном списке, для каждого отчёта изменяем ключ ‘text’ через функцию re.sub, которая ищёт фрагменты текста по шаблону и заменяет их.
- 4) Выводим изменённый список на экран.

1.6 Задание 6

В данном задании необходимо было написать программу, которая формирует список SCP-объектов, которые требуют усиленных мер содержания. На рисунке 6 представлен код программы.

```

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

```

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Дан список SCP-объектов с указанием их класса содержания.
- 2) Через print выводим: список в который с помощью функции filter отбираем все объекты у которых класс выше безопасного.

1.7 Задание 7

В данном задании необходимо было написать программу, которая выводит список трёх самых ресурсоемких инцидентов. На рисунке 7 представлен код программы.

```

1      # Задание 7
2      incidents = [
3          {"id": 101, "staff": 4}, 
4          {"id": 102, "staff": 12}, 
5          {"id": 103, "staff": 7}, 
6          {"id": 104, "staff": 20} 
7      ]
8      pers = sorted(incidents, key=lambda x: x['staff'])
9      print(pers[-3:])

```

Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Дан список инцидентов с количеством задействованного персонала.
- 2) С помощью функции sorted сортирует список по количеству персонала.

3) Далее выводим на экран последний 3 инцидента из списка, то есть 3 самых ресурсоемких инцидента.

1.8 Задание 8

В задании необходимо было написать программу, которая будет создавать новый список строк изменённого вида. На рисунке 8 представлен код программы.

```

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

```

Рисунок 8 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Дан список протоколов безопасности и их уровней критичности.
- 2) Через print выводим: список в который с помощью функции map меняет каждый элемент на вид: "Protocol Lockdown - Criticality 5".

1.9 Задание 9

В этом задании необходимо было написать программу, которая отбирает смены из списка по времени. На рисунке 9 представлен код программы.

```

1 # Задание 9
2 shifts = [6, 12, 8, 24, 10, 4]
3 print(list(filter(lambda x: x >= 8 and x <= 12, shifts)))

```

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Дан список смен охраны с указанием длительности.
- 2) Через print выводим: список в который с помощью функции filter отбираем смены по времени от 8 до 12 включительно.

1.10 Задание 10

В задании необходимо было написать программу, которая выводит имя и результат сотрудника с наивысшей психологической оценкой. На рисунке 10 представлен код программы.

```
1 # Задание 10
2 evaluations = [
3     {"name": "Agent Cole", "score": 78},
4     {"name": "Dr. Weiss", "score": 92},
5     {"name": "Technician Moore", "score": 61},
6     {"name": "Researcher Lin", "score": 88}
7 ]
8 best = max(evaluations, key=lambda x: x['score'])
9 print(best['name'], best['score'])
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Дан список сотрудников с результатами психологической оценки.
- 2) Находим словарь сотрудника с наивысшей оценкой с помощью функции `max` по лямбда-выражению.
- 3) Далее выводим на экран имя и оценку из словаря