

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

Н.Д. Ананко

Ассистент

преподавателя

М.В. Водяницкий

Владивосток 2026

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:

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:

"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):

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

Используя `filter` и лямбда-выражение:

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

Задание 6. Дан список SCP-объектов с указанием их класса содержания: 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:

Отсортируйте инциденты по количеству персонала. 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:

длятся не менее 8 часов 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

Содержание

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

1 Выполнение работы

1.1 Задание 1

В данном задании требуется отсортировать список объектов Фонда 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 print(sorted(objects, key=lambda x: x[1]))

```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Используется встроенная функция sorted
- 2) В качестве ключа сортировки передаётся лямбда-выражение, которое берёт второй элемент кортежа (item[1])
- 3) Результат присваивается переменной objects и выводится на экран

После выполнения программы в консоль выводится отсортированный список объектов, начиная с наименьшего уровня угрозы.

1.2 Задание 2

В данном задании требуется рассчитать общую стоимость работы каждого сотрудника и найти сотрудника с максимальной стоимостью. На рисунке 2 представлен код полученной 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 best = max(staff_shifts, key=lambda x: x["shift_cost"] * x["
9     shifts"])
10 print(f"Максимальная стоимость: {best['name']}: {best['
11     shift_cost'] * best['shifts']}")

```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Используется функция max с ключом - лямбда-выражением, которое вычисляет произведение shift_cost на shifts для каждого сотрудника
- 2) Результат - словарь сотрудника с наибольшей общей стоимостью

3) Выводится имя сотрудника и его максимальная стоимость в форматированной строке

После выполнения программы в консоль выводится имя сотрудника и его максимальная общая стоимость работы.

1.3 Задание 3

В данном задании требуется добавить каждому сотруднику категорию доступа на основе его уровня. На рисунке 3 представлен код полученной программы.

```

1 personnel = [
2     {"name": "Dr. Klein", "clearance": 2},
3     {"name": "Agent Brooks", "clearance": 4},
4     {"name": "Technician Reed", "clearance": 1}
5 ]
6
7 AccessLevels = {1: 'Restricted', 2: 'Confidential', 3: '
    Confidential', 4: 'Top Secret'}
8 print(list(map(lambda p: {**p, "category": AccessLevels[p["
    clearance"]]}, personnel)))

```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Используется map и лямбда-выражение для перебора списка сотрудников
- 2) К каждому словарю сотрудника добавляется новое поле "category" значение которого берётся из словаря AccessLevels по ключу clearance
- 3) Результат преобразуется в список и выводится на экран

После выполнения программы в консоль выводится обновлённый список сотрудников, каждый из которых содержит поле "category" с соответствующей категорией доступа.

1.4 Задание 4

В данном задании требуется отфильтровать зоны, которые работают строго в дневное время (с 8 до 18 часов включительно). На рисунке 4 представлен код полученной программы.

```

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 print(list(filter(lambda z: z["active_from"] <= 8 and z["
    active_to"] >= 18, zones)))

```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Используется функция `filter` с лямбда-выражением
- 2) Условие: `active_from ≤ 8` и `active_to ≥ 18` — то есть зона активна всё время с 8 до 18
- 3) Результат преобразуется в список и выводится на экран

После выполнения программы в консоль выводится список зон, которые полностью покрывают дневной период с 8 до 18 часов.

1.5 Задание 5

В данном задании требуется отфильтровать отчёты, содержащие внешние ссылки, и заменить их на текст [данные удалены]. На рисунке 5 представлен код полученной 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 redacted_reports = filter(lambda report: 'http' in report['
35     text'],reports)
36
37 redacted_reports = list(map(lambda report: {**report,
38     "text": ' '.join('ДААННЫЕ[ УДАЛЕННЫ'] if 'http' in word
39     else word for word in report['text'].split()
40     )},redacted_reports))
41
42 print(redacted_reports)

```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Используется filter с лямбда-выражением, проверяющим наличие "http" или "https" в поле "text"
- 2) Отобранные отчёты преобразуются: в поле "text" ссылки заменяются на строку "[данные удалены]" с помощью map и лямбды
- 3) Результат выводится как список обновлённых отчётов

После выполнения программы в консоль выводится список отчётов, в которых все внешние ссылки заменены на [данные удалены].

1.6 Задание 6

В данном задании требуется отобрать SCP-объекты, которые требуют усиленных мер содержания - то есть все объекты, чей класс не равен "Safe". 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 print(list(filter(lambda s: s["class"] != "Safe",
10                  scp_objects)))

```

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Используется filter с лямбда-выражением, проверяющим условие: scp_object['class'] != 'Safe'
- 2) Результат преобразуется в список и выводится на экран

После выполнения программы в консоль выводится список словарей - всех SCP-объектов, кроме тех, что имеют класс "Safe".

1.7 Задание 7

В данном задании требуется отсортировать инциденты по убыванию количества персонала и оставить только три самых ресурсоёмких. 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 print(sorted(incidents, key=lambda i: i["staff"], reverse=
    True)[:3])

```

Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Используется sorted с лямбда-выражением, сортирующим по значению "staff" в порядке убывания (reverse=True)
- 2) Из отсортированного списка берутся первые три элемента с помощью среза [:3]
- 3) Результат выводится на экран

После выполнения программы в консоль выводится список из трёх инцидентов с наибольшим количеством задействованного персонала.

1.8 Задание 8

В данном задании требуется преобразовать список кортежей с названиями протоколов и их критичностью в список строк в заданном формате. 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 print(list(map(lambda p: f"Protocol {p[0]} - Criticality {p
    [1]}", protocols)))

```

Рисунок 8 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Используется map и лямбда-выражение, которое для каждого кортежа формирует строку: "Protocol название - Criticality критичность"
- 2) Результат преобразуется в список и выводится на экран

После выполнения программы в консоль выводится список строк, каждая из которых содержит название протокола и его уровень критичности в указанном формате.

1.9 Задание 9

В данном задании требуется отфильтровать смены охраны, длительность которых находится в диапазоне от 8 до 12 часов включительно. На рисунке 9 представлен код программы.

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

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:

1) Используется filter с лямбда-выражением, проверяющим условие: shift >= 8 and shift <= 12

2) Результат преобразуется в список и выводится на экран

После выполнения программы в консоль выводится список смен, длина которых составляет от 8 до 12 часов включительно.

1.10 Задание 10

В данном задании требуется найти сотрудника с наивысшей оценкой из списка. 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 best = max(evaluations, key=lambda x: x["score"])
9 print(f"{best['name']}: {best['score']}")
```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

1) Используется функция max с лямбда-выражением, выбирающим элемент с максимальным значением по ключу "score"

2) Выводится имя и балл этого сотрудника в форматированной строке

После выполнения программы в консоль выводится имя сотрудника и его максимальный балл.