

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-25-3

Р.А. Сакс

Ассистент

преподавателя

М.В. Водяницкий

Владивосток 2025

Задание

Выполнить задания на Python с использованием функций высшего порядка (`sorted`, `map`, `filter`, `max`) и лямбда-выражений, оформить отчет по стандартам ВВГУ.

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

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: 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. Используя `filter` и лямбда-выражение, сформируйте список SCP-объектов, которые требуют усиленных мер содержания (класс которых не равен "Safe").

Задание 7. Дан список инцидентов с количеством задействованного персонала. 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. 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). Используя `filter` и лямбда-выражение, выберите только те смены, которые длятся не менее 8 часов и не превышают 12 часов.

Задание 10. Дан список сотрудников с результатами психологической оценки (от 0 до 100). Используя `max` и лямбда-выражение, определите сотрудника с наивысшей оценкой.

Содержание

1	Выполнение работы	3
1.1	Задание 1	3
1.2	Задание 2	4
1.3	Задание 3	5
1.4	Задание 4	6
1.5	Задание 5	7
1.6	Задание 6	8
1.7	Задание 7	9
1.8	Задание 8	10
1.9	Задание 9	11
1.10	Задание 10	12

1 Выполнение работы

1.1 Задание 1

В данном задании был использован метод `sorted` с лямбда-выражением для сортировки списка объектов Фонда по возрастанию уровня угрозы. Лямбда-выражение ключом сортировки задает второй элемент кортежа (уровень угрозы).

```

1 def Lab7Zad1():
2     objects = [
3         ("Containment Cell A", 4),
4         ("Archive Vault", 1),
5         ("Bio Lab Sector", 3),
6         ("Observation Wing", 2)
7     ]
8     SortedObject = sorted(objects, key = lambda x: x[1])
9     print(SortedObject)
10
11 # Lab7Zad1()
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Создан список объектов Фонда в виде кортежей (название, уровень угрозы).
- 2) Использована функция `sorted()` с параметром `key=lambda x: x[1]` для сортировки по второму элементу кортежа.
- 3) Лямбда-выражение `lambda x: x[1]` возвращает уровень угрозы для каждого объекта.
- 4) Результат сортировки выводится на экран.

Результат выполнения:

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

1.2 Задание 2

Для решения задачи был использован `map` с лямбда-выражением, вычисляющим общую стоимость работы каждого сотрудника как произведение стоимости смены на количество смен. Затем с помощью функции `max` была определена максимальная стоимость.

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Создан список сотрудников с информацией о стоимости смены и количестве смен.
- 2) Использована функция `map()` с лямбда-выражением для вычисления общей стоимости работы каждого сотрудника.

3) Лямбда-выражение `lambda x: x["shift_cost"] * x["shifts"]` умножает стоимость смены на количество смен.

4) Функция `max()` находит максимальное значение в полученном списке.

5) Результаты выводятся на экран.

Результат выполнения:

[1800, 1980, 1500]

Max: 1980

1.3 Задание 3

Используя `map` и лямбда-выражение с условной логикой, был создан новый список словарей, в который добавлена категория допуска в зависимости от уровня.

```
1 def Lab7Zad1():
2     objects = [
3         ("Containment Cell A", 4),
4         ("Archive Vault", 1),
5         ("Bio Lab Sector", 3),
6         ("Observation Wing", 2)
7     ]
8     SortedObject = sorted(objects, key = lambda x: x[1])
9     print(SortedObject)
10
11 # Lab7Zad1()
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

1) Создан список персонала с уровнями допуска.

2) Использована функция `map()` с лямбда-выражением для добавления категории допуска.

3) Лямбда-выражение содержит условную логику для определения категории:

- Restricted для уровня 1
- Confidential для уровней 2-3
- Top Secret для уровня 4 и выше

4) Создан новый список словарей с добавленной категорией допуска.

5) Результат выводится на экран.

Результат выполнения:

```
[
{'name': 'Dr. Klein', 'clearance': 2, 'category': 'Confidential'},
{'name': 'Agent Brooks', 'clearance': 4, 'category': 'Top Secret'},
{'name': 'Technician Reed', 'clearance': 1, 'category': 'Restricted'}]
```

]

1.4 Задание 4

С помощью `filter` и лямбда-выражения были отфильтрованы зоны, активность которых начинается не ранее 8:00 и заканчивается не позднее 18:00.

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Создан список зон Фонда с временем активности.
- 2) Использована функция `filter()` с лямбда-выражением для отбора зон.
- 3) Лямбда-выражение `lambda zone: zone["active_from"] >= 8 and zone["active_to"] <= 18` проверяет, что зона работает с 8:00 до 18:00 включительно.
- 4) Отфильтрованные зоны выводятся на экран с указанием времени работы.

Результат выполнения:

(8:00-18:00):

Sector-12: 8:00 - 18:00

1.5 Задание 5

Задача решена в два этапа: сначала `filter` отбирает отчеты, содержащие ссылки, затем с помощью строковых методов выполняется замена ссылок на [ДАННЫЕ УДАЛЕНЫ].

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Создан список отчетов, некоторые из которых содержат ссылки.
- 2) Использована функция `filter()` с лямбда-выражением для отбора отчетов, содержащих `http://` или `https://`.
- 3) Для каждого отчета с ссылками выполняется замена подстрок, содержащих протоколы `http://` или `https://`, на текст [].
- 4) Используются строковые методы `find()` для поиска позиций ссылок и срезы для замены.
- 5) Модифицированные отчеты выводятся на экран.

1.6 Задание 6

Используя `filter` и лямбда-выражение, были отобраны объекты, класс которых не равен "Safe".

Рисунок 6 – Листинг программы для задания 6

Пояснение работы программы:

- 1) Создан список SCP-объектов с классами содержания.
- 2) Использована функция `filter()` с лямбда-выражением для отбора объектов, требующих усиленных мер содержания.
- 3) Лямбда-выражение `lambda obj: obj["class"] != "Safe"` проверяет, что класс объекта не равен "Safe".
- 4) Отфильтрованные объекты выводятся на экран с указанием их кода и класса.

Результат выполнения:

```
SCP-      ,                :
SCP-096 -      : Euclid
SCP-173 -      : Euclid
SCP-055 -      : Keter
SCP-3001 -      : Keter
```

1.7 Задание 7

Сначала инциденты сортируются по количеству персонала в убывающем порядке, затем с помощью среза выбираются три первых (наиболее ресурсоемких).

Рисунок 7 – Листинг программы для задания 7

Пояснение работы программы:

- 1) Создан список инцидентов с количеством задействованного персонала.
- 2) Использована функция `sorted()` с параметрами:
 - `key=lambda x: x["staff"]` для сортировки по количеству персонала
 - `reverse=True` для сортировки по убыванию
- 3) С помощью среза `[:3]` выбраны три первых (наиболее ресурсоемких) инцидента.
- 4) Отсортированный список и три наиболее ресурсоемких инцидента выводятся на экран.

1.8 Задание 8

Метод `map` с лямбда-выражением преобразует кортежи в строки заданного формата.

Рисунок 8 – Листинг программы для задания 8

Пояснение работы программы:

- 1) Создан список протоколов безопасности в виде кортежей (название, уровень критичности).

2) Использована функция `map()` с лямбда-выражением для преобразования кортежей в строки.

3) Лямбда-выражение `lambda protocol: f"Protocol {protocol[0]} - Criticality {protocol[1]}"` создает строку заданного формата.

4) Преобразованные строки выводятся на экран.

Результат выполнения:

```

:
Protocol Lockdown - Criticality 5
Protocol Evacuation - Criticality 4
Protocol Data Wipe - Criticality 3
Protocol Routine Scan - Criticality 1

```

1.9 Задание 9

Лямбда-выражение в `filter` проверяет, находится ли длительность смены в диапазоне от 8 до 12 часов включительно.

```

1 def Lab7Zad1():
2     objects = [
3         ("Containment Cell A", 4),
4         ("Archive Vault", 1),
5         ("Bio Lab Sector", 3),
6         ("Observation Wing", 2)
7     ]
8     SortedObject = sorted(objects, key = lambda x:x[1])
9     print(SortedObject)
10
11 # Lab7Zad1()

```

Рисунок 9 – Листинг программы для задания 9

Пояснение работы программы:

- 1) Создан список длительностей смен охраны.
- 2) Использована функция `filter()` с лямбда-выражением для отбора смен.
- 3) Лямбда-выражение `lambda duration: 8 <= duration <= 12` проверяет, что длительность смены находится в диапазоне от 8 до 12 часов включительно.

4) Исходный список и отфильтрованные смены выводятся на экран.

Результат выполнения:

```

: [6, 12, 8, 24, 10, 4]
8   12           : [12, 8, 10]

```

1.10 Задание 10

Функция `max` с лямбда-выражением, задающим ключ сравнения по полю `"score"` находит сотрудника с наивысшей оценкой.

```

1 def Lab7Zad1():
2     objects = [
3         ("Containment Cell A", 4),
4         ("Archive Vault", 1),
5         ("Bio Lab Sector", 3),
6         ("Observation Wing", 2)
7     ]
8     SortedObject = sorted(objects, key = lambda x: x[1])
9     print(SortedObject)
10
11 # Lab7Zad1()

```

Рисунок 10 – Листинг программы для задания 10

Пояснение работы программы:

- 1) Создан список сотрудников с результатами психологической оценки.
- 2) Использована функция `max()` с параметром `key=lambda emp: emp["score"]` для поиска сотрудника с максимальной оценкой.
- 3) Лямбда-выражение возвращает значение поля `"score"` для каждого сотрудника.
- 4) Найденный сотрудник с наивысшей оценкой выводится на экран с указанием имени и балла.

Результат выполнения:

```
: Dr. Weiss (92    )
```