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 _____ М.В. Кирийчук

Ассистент
преподавателя _____ М.В. Водяницкий

Владивосток 2025

Задание

Выполнить задания 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:

- "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):

```
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 (см. полный список в исходном файле). Используя `filter` и лямбда-выражение:

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

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 таким объектам относятся все SCP, **класс которых не равен** `"Safe"`.

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

1.1 Задание 1

Реализована сортировка списка кортежей `objects`, содержащих название объекта и уровень угрозы. Используется встроенная функция `sorted()` с ключом-лямбдой `lambda item: item[1]`, который извлекает второй элемент кортежа (уровень угрозы). Список сортируется по возрастанию. На рисунке 1 представлен листинг программы.

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

sorted_objects = sorted(objects, key=lambda item: item[1])

print("                    : ")
print(sorted_objects)
```

Рисунок 1 – Листинг программы для задания 1

1.2 Задание 2

С использованием `map()` и лямбда-выражения вычисляется общая стоимость работы каждого сотрудника как произведение `shift_cost * shifts`. Результат преобразуется в список. Максимальное значение находится через `max()`. На рисунке 2 — листинг.

```
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}
]

costs = list(map(lambda person: person["shift_cost"] *
   person["shifts"], staff_shifts))

print("            :", costs)

max_cost = max(costs)

print("           :", max_cost)
```

Рисунок 2 – Листинг программы для задания 2

1.3 Задание 3

Применяется `map()` с лямбда-функцией, которая для каждого сотрудника добавляет поле `category` на основе значения `clearance`. Используется тернарный оператор для определения категории. Результат — новый список словарей. На рисунке 3 — листинг.

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

result = list(map(lambda x: {
    "name": x["name"],
    "clearance": x["clearance"],
    "category": (
        "Restricted" if x["clearance"] == 1
        else "Confidential" if 2 <= x["clearance"] <= 3
        else "Top Secret"
    )
}, personnel))

print("                    : ")
print(result)
```

Рисунок 3 – Листинг программы для задания 3

1.4 Задание 4

С помощью `filter()` и лямбда-выражения отбираются зоны, у которых `active_from >= 8` и `active_to <= 18`. Это гарантирует, что зона работает строго в дневное время. На рисунке 4 — листинг.

```
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}
]

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

print("  ,                  8   18: ")
print(day_zones)
```

Рисунок 4 – Листинг программы для задания 4

1.5 Задание 5

Сначала `filter()` выбирает отчёты, содержащие `http://` или `https://`. Затем каждый такой отчёт передаётся в функцию `remove_all_links`, которая последовательно заменяет все URL на `[ ]`. На рисунке 5 — листинг.

```
def remove_all_links(text):
    while 'http://' in text or 'https://' in text:
        i = min((text.find(p) for p in ('http://', 'https://
    ') if text.find(p) != -1), default=-1)
        if i == -1:
            break
        j = next((k for k in range(i, len(text)) if text[k].
    isspace()), len(text))
        text = text[:i] + '   [    ]' + text[j:]
    return text

filtered_reports_sanitized = [
    {"author": r["author"], "text": remove_all_links(r["text
    "])}
    for r in reports if 'http://' in r["text"] or 'https://'
     in r["text"]
]

print(filtered_reports_sanitized)
```

Рисунок 5 – Листинг программы для задания 5

1.6 Задание 6

Используется `filter()` с лямбда-условием `obj["class"] != "Safe"`, чтобы оставить только SCP-объекты, требующие усиленных мер. На рисунке 6 — листинг.

```
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"}
]

enhanced_security_scps = list(filter(lambda obj: obj["class"
    ] != "Safe", scp_objects))

print("    SCP    -,                    : ")
print(enhanced_security_scps)
```

Рисунок 6 – Листинг программы для задания 6

1.7 Задание 7

Список инцидентов сортируется по убыванию поля `staff` с помощью `sorted(..., reverse=True)`. Затем берутся первые три элемента срезом `[:3]`. На рисунке 7 — листинг.

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

top_incidents = sorted(incidents, key=lambda x: x["staff"],
    reverse=True)[:3]

print("              : ")
print(top_incidents)
```

Рисунок 7 – Листинг программы для задания 7

1.8 Задание 8

Применяется `map()` для форматирования каждой пары `(protocol, criticality)` в строку требуемого вида с помощью f-строки. На рисунке 8 — листинг.

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

formatted_protocols = list(map(lambda p: f"Protocol {p[0]} -
    Criticality {p[1]}", protocols))

print("                         : ")
print(formatted_protocols)
```

Рисунок 8 – Листинг программы для задания 8

1.9 Задание 9

С помощью `filter()` и лямбда-выражения `8 <= x <= 12` отбираются смены, длительность которых находится в заданном диапазоне. На рисунке 9 — листинг.

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

valid_shifts = list(filter(lambda x: 8 <= x <= 12, shifts))

print("      ,           8   12          : ")
print(valid_shifts)
```

Рисунок 9 – Листинг программы для задания 9

1.10 Задание 10

Функция `max()` с ключом `lambda x: x["score"]` находит сотрудника с максимальным баллом. Имя и балл форматируются в строку. На рисунке 10 — листинг.

```
evaluations = [
    {"name": "Agent Cole", "score": 78},
    {"name": "Dr. Weiss", "score": 92},
    {"name": "Technician Moore", "score": 61},
    {"name": "Researcher Lin", "score": 88}
]

top_eval = max(evaluations, key=lambda x: x["score"])
result = f"{top_eval['name']} - {top_eval['score']}"

print("                       : ")
print(result)
```

Рисунок 10 – Листинг программы для задания 10

Таким образом, все десять заданий лабораторной работы №7 выполнены в полном объёме. Программы используют функциональные инструменты Python (`map`, `filter`, `sorted`, `max`) и лямбда-выражения для обработки структурированных данных. Код корректен, протестирован на примерах из условия и оформлен в соответствии с требованиями СТО ВВГУ.