

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 _____ Д.В. Тимкин
Ассистент
преподавателя _____ М.В. Водяницкий

Задание

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

```
reports = [  
    {"author": "Dr. Moss", "text": "Analysis completed. Reference: http://external-archive.net"},  
    {"author": "Agent Lee", "text": "Incident resolved without escalation."},  
    {"author": "Dr. Patel", "text": "Supplementary data available at https://secure-research.org"},  
    {"author": "Supervisor Kane", "text": "No anomalies detected during inspection."},  
    {"author": "Researcher Bloom", "text": "Extended observations uploaded to http://research-notes.lab"},  
    {"author": "Agent Novak", "text": "Perimeter secured. No external interference observed."},  
    {"author": "Dr. Hargreeve", "text": "Full containment log stored at https://internal-db.scp"},  
    {"author": "Technician Moore", "text": "Routine maintenance completed successfully."},  
    {"author": "Dr. Alvarez", "text": "Cross-reference materials: http://crosslink.foundation"},  
    {"author": "Security Officer Tan", "text": "Shift completed without incidents."},  
    {"author": "Analyst Wright", "text": "Statistical model published at https://analysis-hub.org"},  
    {"author": "Dr. Kowalski", "text": "Behavioral deviations documented internally."},  
    {"author": "Agent Fischer", "text": "Additional footage archived: http://video-storage.sec"},  
    {"author": "Senior Researcher Hall", "text": "All test results verified and approved."},  
    {"author": "Operations Lead Grant", "text": "Emergency protocol draft shared via https://ops-share.scp"}]
```

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:

- делятся 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}
```

```
]
```

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.....	7
1.8	Задание 8.....	8
1.9	Задание 9.....	9
1.10	Задание 10.....	10

1 Выполнение работы

1.1 Задание 1

В данном задании был создан список объектов Фонда с указанием уровня угрозы для каждого объекта. С использованием LINQ и лямбда-выражения выполнялась сортировка объектов по возрастанию уровня угрозы, после чего отсортированный список выводился на экран.

```
// Задание 1
var objects = new List<ObjectInfo>
{
    new ObjectInfo("Containment Cell A", 4),
    new ObjectInfo("Archive Vault", 1),
    new ObjectInfo("Bio Lab Sector", 3),
    new ObjectInfo("Observation Wing", 2)
};

var sortedObjects = objects.OrderBy(o => o.ThreatLevel).ToList();
Console.WriteLine("Задание 1:");
foreach (var obj in sortedObjects)
    Console.WriteLine($"{obj.Name} - уровень угрозы {obj.ThreatLevel}");
```

Рисунок 1 – Листинг программы для задания 1

Пояснение работы программы:

- 1) Создается список объектов, содержащих название и уровень угрозы.
- 2) С помощью метода OrderBy и лямбда-выражения выполняется сортировка по уровню угрозы.
- 3) Результат сортировки сохраняется в новый список.
- 4) Выполняется вывод отсортированных объектов на экран.

1.2 Задание 2

В данном задании был создан список сотрудников Фонда с указанием стоимости одной смены и количества проведенных смен. С использованием LINQ вычислялась общая стоимость работы каждого сотрудника, а также определялась максимальная сумма затрат.

```
// Задание 2
var staff = new List<StaffMember>
{
    new StaffMember("Dr. Shaw", 120, 15),
    new StaffMember("Agent Torres", 90, 22),
    new StaffMember("Researcher Hall", 150, 10)
};

var totalCosts = staff.Select(s => s.ShiftCost * s.Shifts).ToList();
int maxCost = totalCosts.Max();

Console.WriteLine("\nЗадание 2:");
foreach (var cost in totalCosts)
    Console.WriteLine(cost);
Console.WriteLine("Максимальная стоимость: " + maxCost);
```

Рисунок 2 – Листинг программы для задания 2

Пояснение работы программы:

- 1) Создается список сотрудников с параметрами стоимости смены и их количества.
- 2) С помощью метода Select вычисляется общая стоимость работы каждого сотрудника.
- 3) Результаты сохраняются в отдельный список.
- 4) С помощью метода Max определяется максимальная стоимость.
- 5) Итоговые значения выводятся на экран.

1.3 Задание 3

В данном задании был создан список персонала Фонда с уровнем допуска. С использованием LINQ каждому сотруднику присваивалась категория допуска на основе его уровня, после чего формировался новый список с результатами.

```
// Задание 3
var personnel = new List<Person>
{
    new Person("Dr. Klein", 2),
    new Person("Agent Brooks", 4),
    new Person("Technician Reed", 1)
};

var categorizedPersonnel = personnel.Select(p =>
    new PersonWithCategory(
        p.Name,
        p.Clearance,
        p.Clearance == 1 ? "Restricted" :
        p.Clearance <= 3 ? "Confidential" :
        "Top Secret"
    )
).ToList();

Console.WriteLine("\nЗадание 3:");
foreach (var p in categorizedPersonnel)
    Console.WriteLine($"{p.Name} - {p.Category}");
```

Рисунок 3 – Листинг программы для задания 3

Пояснение работы программы:

- 1) Создается список персонала с указанием уровня допуска.
- 2) Запускается метод Select для преобразования элементов списка.
- 3) Для каждого сотрудника определяется категория допуска по заданным условиям.
- 4) Создается новый объект с именем, уровнем допуска и категорией.
- 5) Результат сохраняется в новый список и выводится на экран.

1.4 Задание 4

В данном задании был создан список зон Фонда с указанием времени их активности. С использованием LINQ выполнялся отбор зон, которые полностью функционируют в дневной период времени.

```
// Задание 4
var zones = new List<Zone>
{
    new Zone("Sector-12", 8, 18),
    new Zone("Deep Storage", 0, 24),
    new Zone("Research Wing", 9, 17)
};

var dayZones = zones.Where(z => z.ActiveFrom <= 8 && z.ActiveTo >= 18).ToList();

Console.WriteLine("\nЗадание 4:");
foreach (var z in dayZones)
    Console.WriteLine(z.Name);
```

Рисунок 4 – Листинг программы для задания 4

Пояснение работы программы:

- 1) Создается список зон с указанием времени начала и окончания активности.
- 2) С помощью метода Where выполняется фильтрация списка.
- 3) Проверяется, что зона активна в период с 8 до 18 часов включительно.
- 4) Отобранные зоны сохраняются в новый список.
- 5) Названия подходящих зон выводятся на экран.

1.5 Задание 5

В данном задании выполнялась обработка служебных отчетов Фонда.

Программа отбирала отчеты, содержащие внешние ссылки, и заменяла их содержимое на обезличенную информацию перед архивированием.

```
// Задание 5
var reports = new List<Report>
{
    new Report("Dr. Moss", "Analysis completed. Reference: http://external-archive.net"),
    new Report("Agent Lee", "Incident resolved without escalation."),
    new Report("Dr. Patel", "Supplementary data available at https://secure-research.org"),
    new Report("Supervisor Kane", "No anomalies detected during inspection."),
    new Report("Researcher Bloom", "Extended observations uploaded to http://research-notes.lab")
};

var cleanedReports = reports
    .Where(r => r.Text.Contains("http://") || r.Text.Contains("https://"))
    .Select(r => new Report(r.Author, "[ДАННЫЕ УДАЛЕНЫ]"))
    .ToList();

Console.WriteLine("\nЗадание 5:");
foreach (var r in cleanedReports)
    Console.WriteLine($"{r.Author}: {r.Text}");
```

Рисунок 5 – Листинг программы для задания 5

Пояснение работы программы:

- 1) Создается список отчетов с автором и текстом сообщения.
- 2) С помощью метода Where отбираются отчеты, содержащие ссылки.
- 3) Используется метод Select для замены текста отчета.
- 4) Вместо исходного текста устанавливается строка «[ДАННЫЕ УДАЛЕНЫ]».
- 5) Результаты сохраняются в новый список и выводятся на экран.

1.6 Задание 6

В данном задании был создан список SCP-объектов с указанием класса содержания. С использованием LINQ формировался список объектов, требующих усиленных мер содержания.

```
// Задание 6
var scpObjects = new List<SCPObject>
{
    new SCPObject("SCP-096", "Euclid"),
    new SCPObject("SCP-173", "Euclid"),
    new SCPObject("SCP-055", "Keter"),
    new SCPObject("SCP-999", "Safe"),
    new SCPObject("SCP-3001", "Keter")
};

var dangerousScp = scpObjects.Where(s => s.Class != "Safe").ToList();

Console.WriteLine("\nЗадание 6:");
foreach (var s in dangerousScp)
    Console.WriteLine(s.Code);
```

Рисунок 6 — Листинг программы задания 6

Пояснение работы программы:

- 1) Создается список SCP-объектов с указанием класса содержания.
- 2) С помощью метода Where выполняется фильтрация списка.
- 3) Отбираются объекты, класс которых не равен "Safe".
- 4) Отфильтрованные объекты сохраняются в новый список.
- 5) Коды SCP-объектов выводятся на экран.

1.7 Задание 7

В данном задании был создан список инцидентов с количеством задействованного персонала. С использованием LINQ выполнялась сортировка инцидентов и отбор наиболее ресурсоемких из них.

```
// Задание 7
var incidents = new List<Incident>
{
    new Incident(101, 4),
    new Incident(102, 12),
    new Incident(103, 7),
    new Incident(104, 20)
};

var topIncidents = incidents
    .OrderByDescending(i => i.StaffCount)
    .Take(3)
    .ToList();

Console.WriteLine("\nЗадание 7:");
foreach (var i in topIncidents)
    Console.WriteLine($"ID: {i.Id}, персонал: {i.StaffCount}");
```

Рисунок 7 — Листинг программы задания 7

Пояснение работы программы:

- 1) Создается список инцидентов с идентификатором и числом сотрудников.
- 2) С помощью метода OrderByDescending выполняется сортировка по количеству персонала.
- 3) Метод Take используется для выбора трех инцидентов с наибольшими значениями.
- 4) Результаты сохраняются в новый список.
- 5) Информация об инцидентах выводится на экран.

1.8 Задание 8

В данном задании был создан список протоколов безопасности с уровнем критичности. С использованием LINQ формировался новый список строк с отформатированным представлением данных.

```
// Задание 8
var protocols = new List<Protocol>
{
    new Protocol("Lockdown", 5),
    new Protocol("Evacuation", 4),
    new Protocol("Data Wipe", 3),
    new Protocol("Routine Scan", 1)
};

var formattedProtocols = protocols
    .Select(p => $"Protocol {p.Name} - Criticality {p.Level}")
    .ToList();

Console.WriteLine("\nЗадание 8:");
foreach (var p in formattedProtocols)
    Console.WriteLine(p);
```

Рисунок 8 — Листинг программы задания 8

Пояснение работы программы:

1. Создается список протоколов с указанием названия и уровня критичности.
2. С помощью метода Select выполняется преобразование элементов списка.
3. Для каждого протокола формируется строка заданного формата.
4. Результаты сохраняются в новый список.
5. Отформатированные строки выводятся на экран.

1.9 Задание 9

В данном задании был создан список смен охраны с указанием их длительности. С использованием LINQ выполнялся отбор смен, соответствующих заданным временным условиям.

```
// Задание 9
var shifts = new List<int> { 6, 12, 8, 24, 10, 4 };

var validShifts = shifts.Where(s => s >= 8 && s <= 12).ToList();

Console.WriteLine("\nЗадание 9:");
foreach (var s in validShifts)
    Console.WriteLine(s);
```

Рисунок 9 — Листинг программы задания 9

Пояснение работы программы:

1. Создается список длительности смен охраны.
2. С помощью метода Where выполняется фильтрация списка.
3. Отбираются смены продолжительностью не менее 8 часов.
4. Исключаются смены, превышающие 12 часов.
5. Подходящие значения выводятся на экран.

1.10 Задание 10

В данном задании был создан список сотрудников с результатами психологической оценки. С использованием LINQ определялся сотрудник с наивысшим показателем оценки.

```
var evaluations = new List<Evaluation>
{
    new Evaluation("Agent Cole", 78),
    new Evaluation("Dr. Weiss", 92),
    new Evaluation("Technician Moore", 61),
    new Evaluation("Researcher Lin", 88)
};

var bestEmployee = evaluations.OrderByDescending(e => e.Score).First();

Console.WriteLine("\nЗадание 10:");
Console.WriteLine($"{bestEmployee.Name} - {bestEmployee.Score}");
```

Рисунок 10 — Листинг программы задания 10

Пояснение работы программы:

1. Создается список сотрудников с именем и результатом оценки.
2. С помощью метода OrderByDescending выполняется сортировка по значению оценки.
3. Метод First используется для выбора наибольшего значения.
4. Полученные данные сохраняются в переменную.
5. Имя сотрудника и его результат выводятся на экран.

