

Московский государственный технический университет  
им. Н.Э. Баумана

Факультет “Информатика и системы управления”  
Кафедра “Системы обработки информации и управления”



Дисциплина «Парадигмы и конструкции языков программирования»

Отчет по лабораторной работе №7

ГМ

**Выполнил:**  
студент группы ИУ5-36Б  
Германенко И. М.  
**Преподаватель:**  
Гапанюк Ю.Е.

Москва, 2025

### Описание задания:

- Реализовать агента с помощью **MailboxProcessor**;
- Агент должен обрабатывать входящие сообщения и выполнять действия (например, выводить данные в консоль);
- Продемонстрировать реакцию на внешние события.

### Листинг кода:

```
using System;
using System.Collections.Generic;
using System.Linq;

class Employee
{
    public int Id;
    public string Name;
    public int DeptId; // ID отдела (для связи один-ко-многим)
    public override string ToString() => $"Emp: {Name} (id={Id})";
}

class Department
{
    public int Id;
    public string Name;
    public override string ToString() => $"Dept: {Name} (id={Id})";
}

// Класс для связи Многие-ко-Многим
class EmpDept
{
    public int EmpId;
    public int DeptId;
}

class Program
{
    static void Main(string[] args)
    {
        // 1. ДАННЫЕ
        var departments = new List<Department>
        {
            new Department { Id = 1, Name = "IT" },
            new Department { Id = 2, Name = "HR" },
            new Department { Id = 3, Name = "Sales" }
        };

        var employees = new List<Employee>
        {
            new Employee { Id = 1, Name = "Alex", DeptId = 1 },
            new Employee { Id = 2, Name = "Anton", DeptId = 1 },
            new Employee { Id = 3, Name = "Boris", DeptId = 2 },
        }
    }
}
```

```

        new Employee { Id = 4, Name = "Anna", DeptId = 3 },
        new Employee { Id = 5, Name = "Cesar", DeptId = 3 }
    };

    // Связь многие-ко-многим (Сотрудник может быть в нескольких отделах)
    var empDepts = new List<EmpDept>
    {
        new EmpDept { EmpId = 1, DeptId = 1 }, // Alex в IT
        new EmpDept { EmpId = 1, DeptId = 3 }, // Alex еще и в Sales
        new EmpDept { EmpId = 2, DeptId = 1 } // Anton в IT
    };

    Console.WriteLine("--- 1. Список сотрудников и отделов (сортировка по
отделу) ---");
    var q1 = from e in employees
              join d in departments on e.DeptId equals d.Id
              orderby d.Name
              select new { Emp = e.Name, Dept = d.Name };

    foreach (var item in q1) Console.WriteLine($"{item.Dept}: {item.Emp}");

    Console.WriteLine("\n--- 2. Фамилии на букву 'А' ---");
    var q2 = employees.Where(e => e.Name.StartsWith("A"));
    foreach (var e in q2) Console.WriteLine(e.Name);

    Console.WriteLine("\n--- 3. Количество сотрудников в каждом отделе ---");
    var q3 = from e in employees
              group e by e.DeptId into g
              join d in departments on g.Key equals d.Id
              select new { Dept = d.Name, Count = g.Count() };

    foreach (var item in q3) Console.WriteLine($"{item.Dept}: {item.Count}");

    Console.WriteLine("\n--- 4. Отделы, где все сотрудники начинаются на 'А'
---");
    var q4 = from d in departments
              join e in employees on d.Id equals e.DeptId into empGroup
              where empGroup.All(e => e.Name.StartsWith("A"))
              select d.Name;
    // IT (Alex, Anton) подходит. Sales (Anna, Cesar) не подходит.
    foreach (var d in q4) Console.WriteLine(d);

    Console.WriteLine("\n--- 5. Многие-ко-многим (Join через таблицу связи) -
--");
    // Кто работает в IT? (Используем таблицу empDepts)
    var q5 = from ed in empDepts
              join e in employees on ed.EmpId equals e.Id
              join d in departments on ed.DeptId equals d.Id
              select new { Emp = e.Name, Dept = d.Name };

```

```
        foreach (var item in q5) Console.WriteLine($"{item.Emp} works in  
{item.Dept}");  
    }  
}
```

## Скриншоты:

```
• root@ILGERPC:/home/PCPL/Lab6/ReflectionPart# cd /home/PCPL/Lab7  
• root@ILGERPC:/home/PCPL/Lab7# dotnet run  
/home/PCPL/Lab7/Program.cs(16,19): warning CS8618: Non-nullable  
d as nullable. [/home/PCPL/Lab7/Lab7.csproj]  
/home/PCPL/Lab7/Program.cs(8,19): warning CS8618: Non-nullable  
as nullable. [/home/PCPL/Lab7/Lab7.csproj]  
--- 1. Список сотрудников и отделов (сортировка по отделу) ---  
HR: Boris  
IT: Alex  
IT: Anton  
Sales: Anna  
Sales: Cesar  
  
--- 2. Фамилии на букву 'A' ---  
Alex  
Anton  
Anna  
  
--- 3. Количество сотрудников в каждом отделе ---  
IT: 2  
HR: 1  
Sales: 2  
  
--- 4. Отделы, где все сотрудники начинаются на 'A' ---  
IT  
  
--- 5. Многие-ко-многим (Join через таблицу связи) ---  
Alex works in IT  
Alex works in Sales  
Anton works in IT  
○ root@ILGERPC:/home/PCPL/Lab7#
```