# **8 Коллекции, классы-прототипы, обобщения**

Задание 1. Система обработки задач (Queue). Класс Task с полями Id, Title, Priority. Класс TaskManager использует Queue<Task> для управления задачами. Реализовать методы AddTask(), ProcessTask(), GetPendingTasks().

Листинг программы:

using System;

using System.Collections;

namespace E

{ public class Task

{ public int Id { get; set; }

public string Title { get; set; }

public string Priority { get; set; }

public Task(int id, string t, string p)

{ Id = id; Title = t; Priority = p; }

public override string ToString()

{ return $"Id: {Id}; Задача: {Title}; Приоритет: {Priority}."; }

}

public class TaskManager

{ private Queue q = new Queue();

public void AddTask(Task k)

{ q.Enqueue(k);

Console.WriteLine($"Добавлена задача: {k}"); }

public void ProcessTask()

{ if (q.Count > 0)

{ Task k = (Task)q.Dequeue();

Console.WriteLine($"Обработана следующая задача: {k}"); }

else

{ Console.WriteLine("Нет задач для обработки."); }

}

public void GetPendingTasks()

{ if (q.Count > 0)

{ Console.WriteLine("Ожидающие задачи:");

foreach (Task k in q)

{ Console.WriteLine(k); }

}

else

{ Console.WriteLine("Все задачи выполнены."); } }

}

class Q

{ static void Main(string[] args)

{ TaskManager m = new TaskManager();

m.AddTask(new Task(1, "Сделать практические задания", "Высокий"));

m.AddTask(new Task(2, "Поспать", "Средний"));

m.AddTask(new Task(3, "Выжить", "Низкий"));

Console.WriteLine(); m.GetPendingTasks();

Console.WriteLine(); m.ProcessTask();

m.ProcessTask(); Console.WriteLine();

m.GetPendingTasks(); } }}

Таблица 8.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Задачи 1, 2, 3 | Задача 3 |

Анализ результатов:

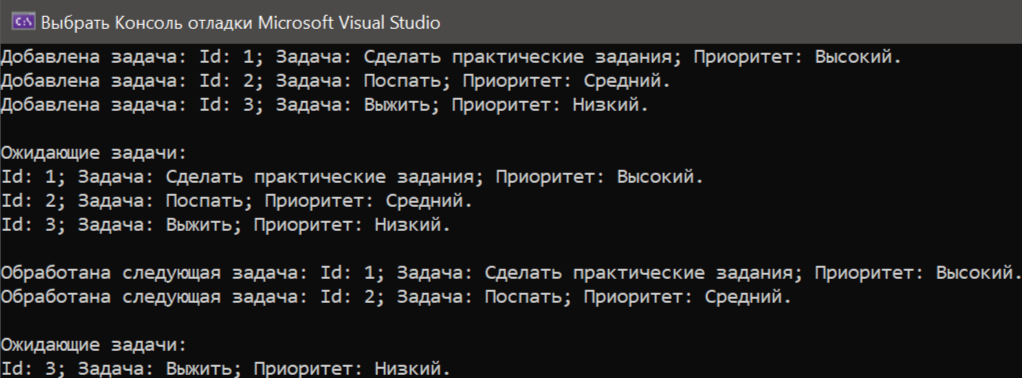


Рисунок 8.1 – Результат работы программы