

Экзаменационное задание. Платформа Microsoft .NET и язык программирования C#

Created пятница 25 Июль 2025

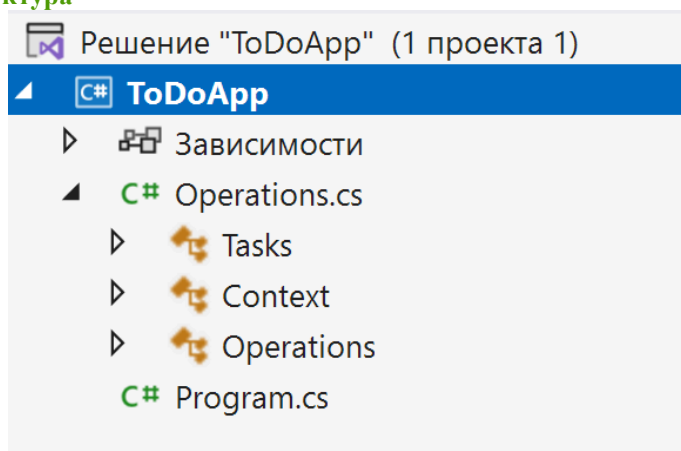
Третье задание

ToDoList - это задачник (список дел)

Условия выполнения:

1. Добавление задач в список дел.
2. Хранение задач (задачи должны сохраняться после закрытия приложения)
3. Отображения всех задач.
4. Поиск определенной задачи (например по словам заголовка или по тексту самой задачи)
5. Сортировка по важности (не важно / Важно / Очень Важно).
6. Сортировка по разным критериям (дата / важность).
7. Удаление задач из списка.
8. Редактирование задачи.

Структура



Код Operations.cs

```
using System;
using System.Linq;
using Microsoft.EntityFrameworkCore;

namespace ToDoApp
{
    // Модель задачи
    public class Tasks
    {
        public int Id { get; set; }
        public string Category { get; set; } = string.Empty;
        public int Priority { get; set; }
        public string Description { get; set; } = string.Empty;
        public string DateTime { get; set; } = string.Empty;
    }

    // Контекст базы данных
    public class Context : DbContext
    {
        public DbSet<Tasks> Tasks { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder options)
            => options.UseSqlite("Data Source=ToDoApp.db");
    }

    // Операции с задачами
    public class Operations
    {
        private readonly Context db;

        public Operations(Context context)
        {
            db = context;
        }

        public void DisplayByDueDate()
        {

```

```

var tasks = db.Tasks.OrderBy(t => t.DateTime).ToList();
if (!tasks.Any())
{
    Console.WriteLine("Список дел пуст");
    return;
}
Console.WriteLine("Список дел по сроку исполнения:");
for (int i = 0; i < tasks.Count; i++)
{
    var t = tasks[i];
    Console.WriteLine($"{i + 1}. (Срок: {t.DateTime}) - {t.Category}: {t.Description} (Приоритет: {t.Priority})");
}
}

public void DisplayByPriority()
{
    var tasks = db.Tasks.OrderBy(t => t.Priority).ToList();
    if (!tasks.Any())
    {
        Console.WriteLine("Список дел пуст");
        return;
    }
    Console.WriteLine("Список дел по приоритету:");
    for (int i = 0; i < tasks.Count; i++)
    {
        var t = tasks[i];
        Console.WriteLine($"{i + 1}. (Приоритет: {t.Priority}) - {t.Category}: {t.Description} (Срок: {t.DateTime})");
    }
}

public void AddTask()
{
    var task = new Tasks();

    Console.Write("Введите категорию задачи: ");
    task.Category = Console.ReadLine() ?? "";

    while (true)
    {
        Console.Write("Введите приоритет (1 - высокий, 2 - средний, 3 - низкий): ");
        if (int.TryParse(Console.ReadLine(), out int prio) && prio >= 1 && prio <= 3)
        {
            task.Priority = prio;
            break;
        }
        Console.WriteLine("Ошибка: введите число от 1 до 3");
    }

    Console.Write("Введите описание задачи: ");
    task.Description = Console.ReadLine() ?? "";

    Console.Write("Введите срок исполнения (гггг-мм-дд чч:мм): ");
    task.DateTime = Console.ReadLine() ?? "";

    db.Tasks.Add(task);
    Console.WriteLine("Задача добавлена! Не забудьте сохранить изменения (пункт 6).");
}

public void RemoveTask()
{
    var tasks = db.Tasks.ToList();
    if (!tasks.Any())
    {
        Console.WriteLine("Нет задач для удаления.");
        return;
    }

    Console.Write("Введите номер задачи для удаления (начиная с 1): ");
    if (int.TryParse(Console.ReadLine(), out int index) && index > 0 && index <= tasks.Count)
    {
        var taskToRemove = tasks[index - 1];
        db.Tasks.Remove(taskToRemove);
        Console.WriteLine("Задача удалена! Не забудьте сохранить изменения (пункт 6).");
    }
    else
    {
        Console.WriteLine("Ошибка: неверный номер задачи!");
    }
}

```

```

}

public void EditTask()
{
    var tasks = db.Tasks.ToList();
    if (!tasks.Any())
    {
        Console.WriteLine("Нет задач для редактирования.");
        return;
    }

    Console.Write("Введите номер задачи для редактирования (начиная с 1): ");
    if (int.TryParse(Console.ReadLine(), out int index) && index > 0 && index <= tasks.Count)
    {
        var task = tasks[index - 1];
        Console.WriteLine($"Редактирование задачи: {task.Category} - {task.Description}");

        Console.Write("Введите новую категорию (оставьте пустым, чтобы не менять): ");
        string newCategory = Console.ReadLine() ?? "";
        if (!string.IsNullOrEmpty(newCategory))
            task.Category = newCategory;

        Console.Write("Введите новый приоритет (1 - высокий, 2 - средний, 3 - низкий, оставьте пустым, чтобы не менять): ");
        string newPriority = Console.ReadLine() ?? "";
        if (int.TryParse(newPriority, out int prio) && prio >= 1 && prio <= 3)
            task.Priority = prio;

        Console.Write("Введите новое описание (оставьте пустым, чтобы не менять): ");
        string newDescription = Console.ReadLine() ?? "";
        if (!string.IsNullOrEmpty(newDescription))
            task.Description = newDescription;

        Console.Write("Введите новый срок исполнения (гггг-мм-дд чч:мм, оставьте пустым, чтобы не менять): ");
        string newDateTime = Console.ReadLine() ?? "";
        if (!string.IsNullOrEmpty(newDateTime))
            task.DateTime = newDateTime;

        db.Tasks.Update(task);
        Console.WriteLine("Задача обновлена! Не забудьте сохранить изменения (пункт 6).");
    }
    else
    {
        Console.WriteLine("Ошибка: неверный номер задачи!");
    }
}

public void SaveChanges()
{
    try
    {
        db.SaveChanges();
        Console.WriteLine("Изменения сохранены.");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Ошибка при сохранении: " + ex.Message);
    }
}

public void CategorySearch()
{
    Console.Write("Введите категорию для поиска: ");
    string category = Console.ReadLine() ?? "";

    var found = db.Tasks.Where(t => t.Category.Equals(category,
        StringComparison.OrdinalIgnoreCase)).ToList();

    if (!found.Any())
    {
        Console.WriteLine("Задачи с такой категорией не найдены.");
        return;
    }

    Console.WriteLine($"Задачи в категории \"{category}\"");
    int i = 1;
    foreach (var task in found)
    {

```

```

        Console.WriteLine($"{i}. {task.Description} (Приоритет: {task.Priority}, Срок: {task.DateTime})");
        i++;
    }
}

public void PrioritySearch()
{
    while (true)
    {
        Console.Write("Введите приоритет для поиска (1 - высокий, 2 - средний, 3 - низкий): ");
        if (int.TryParse(Console.ReadLine(), out int prio) && prio >= 1 && prio <= 3)
        {
            var found = db.Tasks.Where(t => t.Priority == prio).ToList();

            if (!found.Any())
            {
                Console.WriteLine("Задачи с таким приоритетом не найдены.");
                return;
            }

            Console.WriteLine($"Задачи с приоритетом {prio}:");
            int i = 1;
            foreach (var task in found)
            {
                Console.WriteLine($"{i}. {task.Category}: {task.Description} (Срок: {task.DateTime})");
                i++;
            }
            break;
        }
        Console.WriteLine("Ошибка: введите число от 1 до 3");
    }
}
}
}
}
}

```

Код Program.cs

```

using System;
using Microsoft.EntityFrameworkCore;

using ToDoApp;

using var db = new Context();
db.Database.EnsureCreated();

var ops = new Operations(db);

Console.OutputEncoding = System.Text.Encoding.UTF8;
Console.InputEncoding = System.Text.Encoding.UTF8;

while (true)
{
    Console.WriteLine("\nСписок дел:");
    Console.WriteLine("1. Все задачи по сроку исполнения");
    Console.WriteLine("2. Все задачи по приоритету");
    Console.WriteLine("3. Добавить задачу");
    Console.WriteLine("4. Удалить задачу");
    Console.WriteLine("5. Редактировать задачу");
    Console.WriteLine("6. Сохранить изменения");
    Console.WriteLine("7. Искать по категории");
    Console.WriteLine("8. Искать по приоритету");
    Console.WriteLine("9. Выход");
    Console.Write("Выбор: ");

    if (!int.TryParse(Console.ReadLine(), out int choice))
    {
        Console.WriteLine("Ошибка: неверный ввод!");
        continue;
    }

    switch (choice)
    {
        case 1: ops.DisplayByDueDate(); break;
        case 2: ops.DisplayByPriority(); break;
        case 3: ops.AddTask(); break;
        case 4: ops.RemoveTask(); break;
        case 5: ops.EditTask(); break;
        case 6: ops.SaveChanges(); break;
        case 7: ops.CategorySearch(); break;
        case 8: ops.PrioritySearch(); break;
    }
}

```

```
        case 9:
            Console.WriteLine("Выход");
            return;
        default:
            Console.WriteLine("Ошибка: неверный выбор!");
            break;
    }
}
```

Хранилище SQLite

```
CREATE TABLE "tasks" (
    "Id" INTEGER,
    "Category" TEXT,
    "Priority" INTEGER,
    "Description" TEXT,
    "DateTime" TEXT,
    PRIMARY KEY("Id")
)
```

Таблицы (1)

tasksCREATE TABLE "tasks" ("Id" INTEGER, "Category" TEXT, "P

Индексы (0)

Представления (0)

Триггеры (0)

	<u>Id</u>	Category	Priority	Description	DateTime
	Фи...	Фильтр	Фильтр	Фильтр	Фильтр
1	1	Экзамен	1	Закончить экзаменационный проект по...	2024-06-06 15:30
2	2	Досуг	1	Сходить в Практику by Darwin, Ленком	2024-06-09 18:00
3	3	финансы	1	Сделать перевод в бухгалтерию	2024-06-10 20:00
4	4	Экзамен	1	Сделать скрины отладки кода	2024-06-12 16:00
5	5	Купить	2	Беспроводную мышку и проектор	2024-06-05 10:00
6	6	Расписание	2	Посмотреть электронный журнал	2024-06-08 09:00
7	7	Ремонт	2	Обжать кабеля	2024-06-11 14:00
8	8	финансы	2	Обсудить свои счета с менеджером в ...	2024-06-13 10:00
9	9	Досуг	3	Найти ролики на Ютубе по музыкально...	2024-06-14 21:00

Вывод в консоль

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 2

Список дел по приоритету:

1. (Приоритет: 1) - Экзамен: Закончить экзаменационный проект по ООП на шарпе (Срок: 2024-06-06 15:30)
2. (Приоритет: 1) - Досуг: Сходить в Практику by Darwin, Ленком (Срок: 2024-06-09 18:00)
3. (Приоритет: 1) - Финансы: Сделать перевод в бухгалтерию (Срок: 2024-06-10 20:00)
4. (Приоритет: 1) - Экзамен: Сделать скрины отладки кода (Срок: 2024-06-12 16:00)
5. (Приоритет: 2) - Купить: Беспроводную мышку и проектор (Срок: 2024-06-05 10:00)
6. (Приоритет: 2) - Расписание: Посмотреть электронный журнал (Срок: 2024-06-08 09:00)
7. (Приоритет: 2) - Ремонт: Обжать кабеля (Срок: 2024-06-11 14:00)
8. (Приоритет: 2) - Финансы: Обсудить свои счета с менеджером в Сбере (Срок: 2024-06-13 10:00)
9. (Приоритет: 3) - Досуг: Найти ролики на Ютубе по музыкальной теории и посмотреть (Срок: 2024-06-14 21:00)

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 4

Введите номер задачи для удаления (начиная с 1): 2

Задача удалена! Не забудьте сохранить изменения (пункт 6).

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 4

Введите номер задачи для удаления (начиная с 1): 4

Задача удалена! Не забудьте сохранить изменения (пункт 6).

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 6

Изменения сохранены.

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 1

Список дел по сроку исполнения:

1. (Срок: 2024-06-05 10:00) - Купить: Беспроводную мышку и проектор (Приоритет: 2)
2. (Срок: 2024-06-06 15:30) - Экзамен: Закончить экзаменационный проект по ООП на шарпе (Приоритет: 1)
3. (Срок: 2024-06-08 09:00) - Расписание: Посмотреть электронный журнал (Приоритет: 2)
4. (Срок: 2024-06-10 20:00) - Финансы: Сделать перевод в бухгалтерию (Приоритет: 1)
5. (Срок: 2024-06-11 14:00) - Ремонт: Обжать кабеля (Приоритет: 2)
6. (Срок: 2024-06-13 10:00) - Финансы: Обсудить свои счета с менеджером в Сбере (Приоритет: 2)
7. (Срок: 2024-06-14 21:00) - Досуг: Найти ролики на Ютубе по музыкальной теории и посмотреть (Приоритет: 3)

Список дел:

Список дел:

1. Все задачи по сроку исполнения
2. Все задачи по приоритету
3. Добавить задачу
4. Удалить задачу
5. Редактировать задачу
6. Сохранить изменения
7. Искать по категории
8. Искать по приоритету
9. Выход

Выбор: 9

Выход

C:\Users\user\source\repos\ToDoApp\bin\Debug\net9.0\ToDoApp.exe (процесс 41976) завершил работу с кодом 0 (0x0).

Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".

Нажмите любую клавишу, чтобы закрыть это окно...