|  |  |  |
| --- | --- | --- |
| Защищено:  Гапанюк Ю.Е.    "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_**2023** г. |  | Демонстрация:  Тихонова Д.Д.  "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_**2023** г. |

**Отчет по домашнему заданию по курсу**

**Парадигмы и конструкции языков программирования**

#### Тема работы: «Знакомство с языком С#»

6

(количество листов)

Вариант № **24**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы ИУ5Ц-54Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Тихонова Д.Д. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_**2023**  г. |

Москва, МГТУ - **2023**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**СОДЕРЖАНИЕ**

[1. Описание задания 3](#_Toc152715356)

[2. Введение 3](#_Toc152715357)

[3. Объектно-ориентированное программирование в C# 3](#_Toc152715358)

[3. Работа с коллекциями 4](#_Toc152715359)

[4. Проект на С#: Создание простой системы управления задачами (Task Management System) 4](#_Toc152715360)

[5. Заключение 6](#_Toc152715361)

# Описание задания

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

# Введение

C# (C Sharp) – это объектно-ориентированный язык программирования, разработанный компанией Microsoft. Он является частью .NET Framework и предоставляет широкие возможности для разработки приложений на платформе Windows.

# 3. ****Объектно-ориентированное программирование в C#****

Одной из ключевых особенностей C# является поддержка объектно-ориентированного программирования (ООП). В языке присутствуют все основные принципы ООП: наследование, инкапсуляция и полиморфизм.

Пример кода:  
// Пример класса с использованием наследования

class Animal

{

    public void Eat()

    {

        Console.WriteLine("Animal is eating.");

    }

}

class Dog : Animal

{

    public void Bark()

    {

        Console.WriteLine("Dog is barking.");

    }

}

// Пример использования полиморфизма

Animal myDog = new Dog();

myDog.Eat();  // Вызывается метод из класса Animal, несмотря на тип переменной

# Работа с коллекциями

В C# существует богатый набор коллекций, предоставляющих различные структуры данных. Некоторые из них включают List, Dictionary, HashSet и другие. Коллекции позволяют эффективно хранить и управлять группами объектов.

Пример кода:

// Пример использования List для хранения и обработки данных

List<int> numbers = new List<int> { 1, 2, 3, 4, 5 };

numbers.Add(6); // Добавление элемента в список

foreach (int num in numbers)

{

    Console.WriteLine(num);

}

// Пример использования Dictionary для хранения пар ключ-значение

Dictionary<string, int> ages = new Dictionary<string, int>();

ages.Add("John", 25);

ages.Add("Jane", 30);

Console.WriteLine("Age of John: " + ages["John"]);

# Проект на С#: ****Создание простой системы управления задачами (Task Management System)****

**Код проекта:**

Класс Task:

class Task

{

    public string Title { get; set; }

    public string Description { get; set; }

    public DateTime DueDate { get; set; }

    public bool IsCompleted { get; set; }

}

*Класс TaskManager:*

class TaskManager

{

    private List<Task> tasks = new List<Task>();

    public void AddTask(Task task)

    {

        tasks.Add(task);

    }

    public void DisplayTasks()

    {

        foreach (var task in tasks)

        {

            Console.WriteLine($"Title: {task.Title}, Due Date: {task.DueDate}, Completed: {task.IsCompleted}");

        }

    }

    public void MarkTaskAsCompleted(string title)

    {

        var task = tasks.Find(t => t.Title.Equals(title, StringComparison.OrdinalIgnoreCase));

        if (task != null)

        {

            task.IsCompleted = true;

        }

    }

    public void RemoveTask(string title)

    {

        tasks.RemoveAll(t => t.Title.Equals(title, StringComparison.OrdinalIgnoreCase));

    }

}

Пример использования:

class Program

{

    static void Main()

    {

        TaskManager taskManager = new TaskManager();

        // Добавление задач

        taskManager.AddTask(new Task { Title = "Complete Project", Description = "Finish the coding project", DueDate = DateTime.Now.AddDays(7) });

        taskManager.AddTask(new Task { Title = "Meeting", Description = "Attend the team meeting", DueDate = DateTime.Now.AddDays(2) });

        // Просмотр задач

        taskManager.DisplayTasks();

        // Завершение задачи

        taskManager.MarkTaskAsCompleted("Complete Project");

        // Просмотр обновленного списка задач

        taskManager.DisplayTasks();

    }

}

Проект демонстрирует пример использования языка C# для создания простой системы управления задачами. В нём реализованы основные принципы объектно-ориентированного программирования, такие как создание классов и их взаимодействие. Проект также включает работу с коллекциями для управления списком задач.

# Заключение

C# предоставляет разработчикам мощные инструменты для реализации объектно-ориентированных концепций и эффективной работы с коллекциями данных. Эти возможности делают язык популярным среди разработчиков, создающих разнообразные приложения для платформы Windows. Ознакомление с основными конструкциями и парадигмой языка позволяет создавать надежные и эффективные программы.