Министерство образования и науки РФ

Федеральное государственное автономное образовательное учреждение высшего образования

«Омский государственный технический университет»

|  |  |
| --- | --- |
| Факультет (институт) | *Информационных технологий и компьютерных систем* |
|  |  |
| Кафедра | *Прикладная математика и фундаментальная информатика* |
|  |  |

**Расчетно–графическая работа**

|  |  |
| --- | --- |
| по дисциплине | ***Алгоритмизация и программирование*** |
|  |  |
| на тему | Программная реализация задач |

Пояснительная записка

|  |  |
| --- | --- |
| **Шифр проекта** | 020–РГР–02.03.02–№ 08 – ПЗ |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **Студента** | | Киснер Анастасии Евгеньевны | | | | | |
|  |  |  |  | | фамилия, имя, отчество полностью | | | | | |
|  |  |  | Курс | 1 |  | Группа | | МО–241 | | |
|  |  |  |  |  |  | |  |  | |  |
|  | | | **Направление (специальность)** | | | | | ***02.03.03*** | | |
|  | | | Математическое обеспечение и администрирование информационных систем | | | | | | | |
|  |  |  | код, наименование | | | | | | | |
|  |  |  | Руководитель | | ***ст. преподаватель*** | | | | | |
|  |  |  | ученая степень, звание | | | | | |
|  |  |  | ***Федотова И.В.*** | | | | | | | |
|  |  |  | фамилия, инициалы | | | | | | | |
|  |  |  | Выполнил | |  | | | | | |
|  |  |  | дата, подпись студента | | | | | |
|  |  |  |  | | | | | | | |
|  |  |  | **Работа защищена с количеством баллов** | | | | | |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | дата, подпись руководителя |  |  |  |

Омск 2025

**Оглавление**

[Введение 3](#_Toc197294130)

[Постановка задачи «Прямоугольник» 4](#_Toc197294131)

[Ход решения задачи «Прямоугольник» 6](#_Toc197294132)

[Постановка задачи «Города» 8](#_Toc197294133)

[Ход задачи «Города» 10](#_Toc197294134)

[Постановка задачи «Минное поле» 12](#_Toc197294135)

[Ход задачи «Минное поле» 13](#_Toc197294136)

[Постановка задачи «Расстояние между городами» 15](#_Toc197294137)

[Ход задачи «Расстояние между городами» 16](#_Toc197294138)

[Заключение 19](#_Toc197294139)

[Список используемой литературы 20](#_Toc197294140)

# **Введение**

C# – это мощный и универсальный объектно-ориентированный язык программирования от Microsoft, тесно интегрированный с платформой .NET. Его компилируемый характер обеспечивает высокую производительность, а использование промежуточного языка (IL) и среды выполнения .NET (CLR) гарантирует переносимость на различные платформы. Автоматическое управление памятью (сборщик мусора) упрощает разработку и повышает надежность, минимизируя риск утечек памяти. Строгая типизация способствует предотвращению ошибок на этапе компиляции. C# эффективно поддерживает многопоточность, что позволяет создавать высокопроизводительные приложения, а обширная стандартная библиотека .NET предоставляет широкий спектр готовых компонентов для решения разнообразных задач. Он поддерживает различные парадигмы программирования, включая объектно-ориентированное, функциональное (с помощью LINQ) и аспектно-ориентированное программирование. C# широко используется для создания веб-приложений (ASP.NET Core), десктопных приложений (WinForms, WPF, .NET MAUI), мобильных приложений (Xamarin), игр (Unity) и бэкенд-систем, что делает его одним из наиболее востребованных языков программирования в мире. Его популярность обусловлена сочетанием высокой производительности, относительно простого синтаксиса, обширной поддержки со стороны Microsoft и активного сообщества разработчиков, постоянно расширяющего его возможности и обеспечивающего обилие ресурсов для обучения и поддержки. Постоянное развитие языка, добавление новых функций и улучшений гарантирует его актуальность и востребованность в будущем.

# **Постановка задачи «Прямоугольник»**

Имеется прямоугольная область размером N×M клеток, некоторые из которых закрашены.

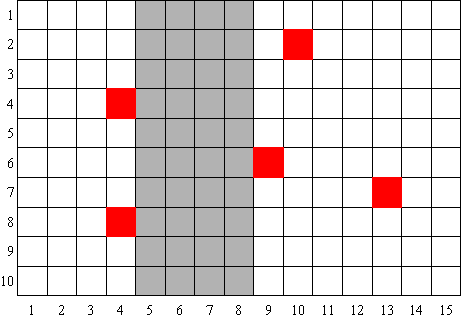
Требуется найти прямоугольник максимальной площади, не содержащий закрашенные клетки. Стороны прямоугольника должны быть параллельны краям прямоугольной области.

Входной файл содержит:

* в первой строке два числа N и M – число столбцов и строк прямоугольной области соответственно (1 ≤ N, M ≤ 250);
* во второй строке число K – количество закрашенных клеток (0 ≤ K ≤ (N\*M-1));
* следующие K строк содержат по 2 числа X и Y – координаты (столбец и строка соответственно) закрашенной клетки.

Выходной файл должен содержать площадь найденного прямоугольника.

Пример:



Красным показаны закрашенные точки, серым – искомый прямоугольник.

Input.txt

15 10

5

10 2

4 4

9 6

13 7

4 8

Output.txt

40

# **Ход решения задачи «Прямоугольник»**

Программа вычисляет максимальную площадь прямоугольника из свободных клеток на прямоугольной сетке.

1. Ввод данных: чтение всех строк входного файла: N – число столбцов, M – число строк.
2. Создание сетки: создаётся двумерный массив, представляющий поле. Все клетки инициализируются как свободные (1).
3. Закрашивание занятых клеток.
4. Подготовка к поиску максимального прямоугольника: heights[j] будет означать количество подряд идущих единиц сверху вниз в j-й колонке.
5. Вывод результата: в файл выводится максимальная площадь прямоугольника из свободных клеток.

Код программы «Прямоугольник» представлен на рисунках 1 и 2:

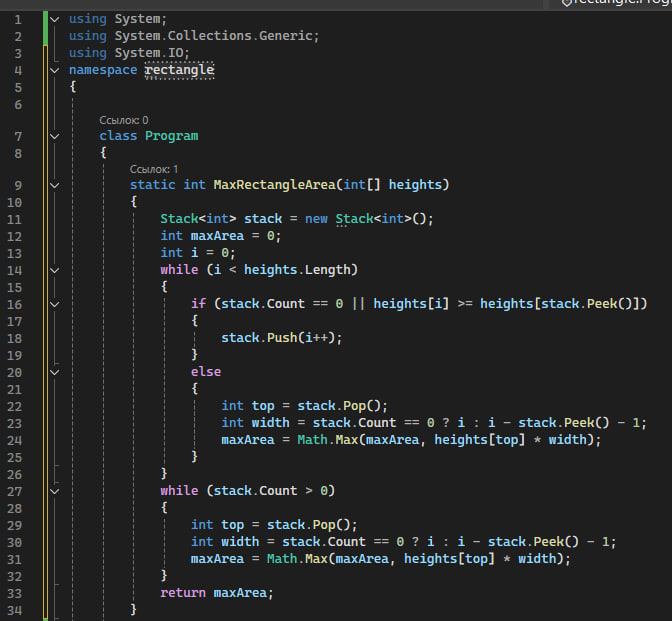


Рисунок 1 – Первая часть кода программы «Прямоугольник»



Рисунок 2 – Вторая часть кода программы «Прямоугольник»

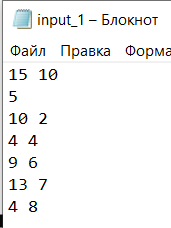


Рисунок 3 – Пример ввода программы «Прямоугольник»

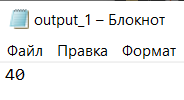


Рисунок 4 – Пример вывода программы «Прямоугольник»

# **Постановка задачи «Города»**

В некоторой стране население живет в городах. Города соединены дорогами. Каждая дорога начинается в одном городе и заканчивается в другом. Длины всех дорог известны. Из любого города через систему дорог можно добраться до любого другого города.

Будем называть расстоянием между двумя городами длину кратчайшего пути между ними.

Требуется найти наибольшее расстояние между городами в стране.

Входной файл

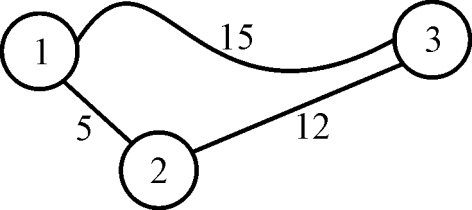
Первая строка содержит два целых числа N и M, разделенных пробелом - количество городов и дорог (2  N  400, 1  M  100000).

Следующие M строк содержат по три целых числа I, J, L - номера городов, соединенных дорогой (1  I, J  N) и длина дороги (1  L  1000).

Выходной файл

Содержит единственное целое число - наибольшее расстояние между городами в стране.

Пример 1:



Input.txt

3 3

1 2 5

3 1 15

2 3 12

Output.txt

15

Пример 2:

Input.txt

4 6

1 2 3

2 3 14

3 4 7

4 1 12

1 3 20

2 4 5

Output.txt

15

# **Ход задачи «Города»**

Этот код реализует поиск кратчайшего пути в неориентированном графе между вершинами 1 и 3 с помощью алгоритма Дейкстры, и сохраняет вес минимального пути в файл.

1. Ввод данных: чтение всех строк входного файла. Первая строка содержит два числа: n (количество городов) и m (количество дорог).
2. Инициализация графа: массив списков смежности graph для представления графа. Используется n+1, так как вершины нумеруются с 1.
3. Заполнение графа: каждая строка – ребро между вершинами u и v с весом w. Добавляется в оба направления, так как дороги двусторонние.
4. Настройка начальной и целевой вершины: начинаем поиск кратчайшего пути из вершины 1 в вершину 3.
5. Алгоритм Дейкстры.
6. Обход соседей и обновление расстояний: рассматриваем всех соседей v вершины u. Если найден более короткий путь – обновляем расстояние и добавляем v в очередь.
7. Вывод результата: в файл записывается минимальный вес пути от вершины 1 до 3.

Код программы «Города» представлен на рисунках 5 и 6:

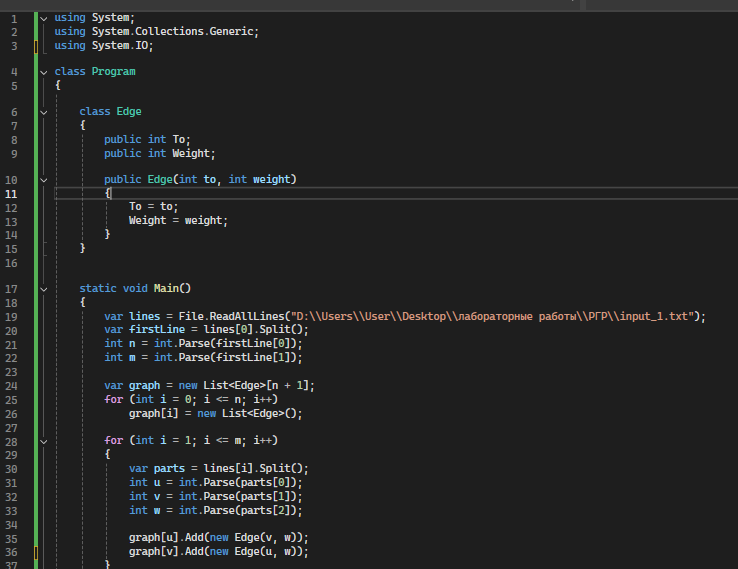


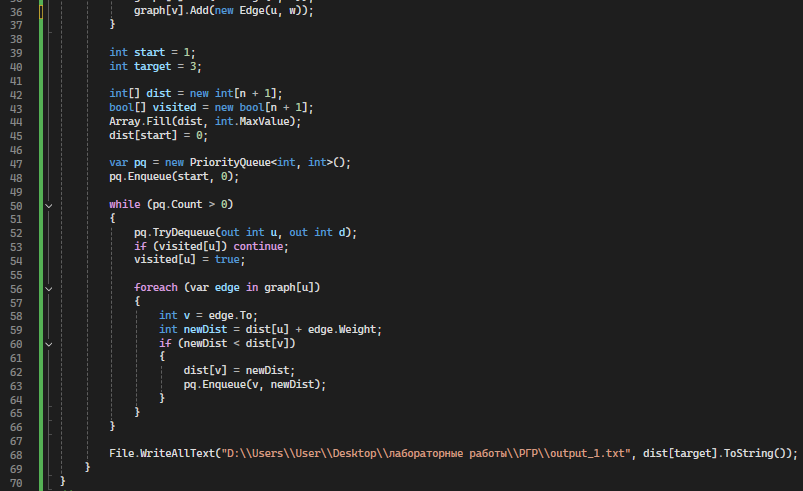
Рисунок 5 – Первая часть кода программы «Города»  


Рисунок 6 – Вторая часть кода программы «Города»

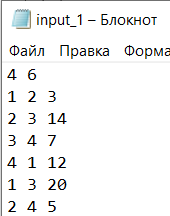


Рисунок 7 – Пример ввода программы «Города»

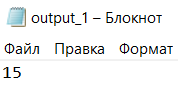
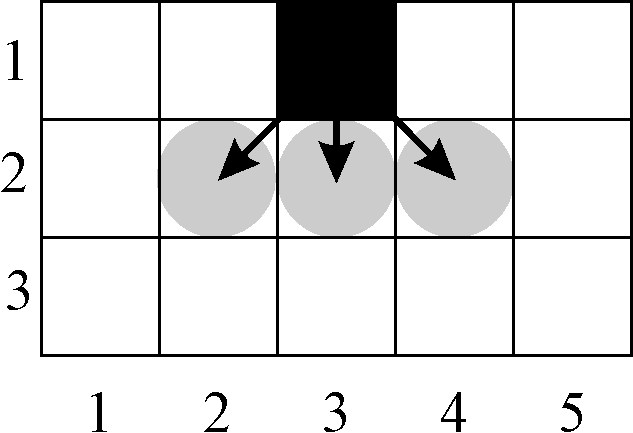


Рисунок 8 – Пример вывода программы «Города»

# **Постановка задачи «Минное поле»**

Минное поле представляет собой прямоугольник, разделенный на квадратные зоны одинакового размера. С помощью средств современной разведки было выяснено, сколько мин заложено в каждой зоне. Для разминирования каждой мины требуется одинаковое время. Для прохода по зоне, необходимо разминировать все мины на ней.

Проход между зонами возможен в следующих направлениях:



т.е. в следующий ряд прямо и по диагоналям.

Требуется найти один из вариантов прохода в минном поле, требующий минимального времени для разминирования.

Примечания:

размер поля N\*M, 0 < N  200, 0 < M  400;

количество мин в каждой зоне от 0 до 100.

Входной файл содержит:

в первой строке числа N и M, разделенные пробелом (количество рядов на минном поле и зон в ряде соответственно);

в следующих N строках содержится по M целых чисел разделенных пробелом – количество мин в зонах i-го ряда (1  i  N).

Выходной файл должен содержать N строк. В каждой строке указывается номер зоны в i-м ряду (1  i  N), через которую лежит найденный проход.

Пример 1:

Input.txt

3 5

5 3 0 8 5

5 3 6 5 4

7 0 0 5 3

Output.txt

3

2

2

# **Ход задачи «Минное поле»**

Программа находит путь с минимальной суммой мин от верхнего до нижнего ряда по правилу прохода: вниз, вниз-влево, вниз-вправо.

1. Ввод данных: считывание всех строк из входного файла input\_1.txt.
2. Инициализация массивов.
3. Считывание мин: заполнение массива mines числами мин из входного файла.
4. Инициализация начала пути
5. Нахождение минимального пути в последнем ряду: находим клетку в последнем ряду с минимальной стоимостью пути.
6. Восстановление пути: обратным проходом восстанавливаем путь.
7. Вывод результата: в файл записывается номер столбца в каждой строке пути.

Код программы «Минное поле» представлен на рисунках 9 и 10:

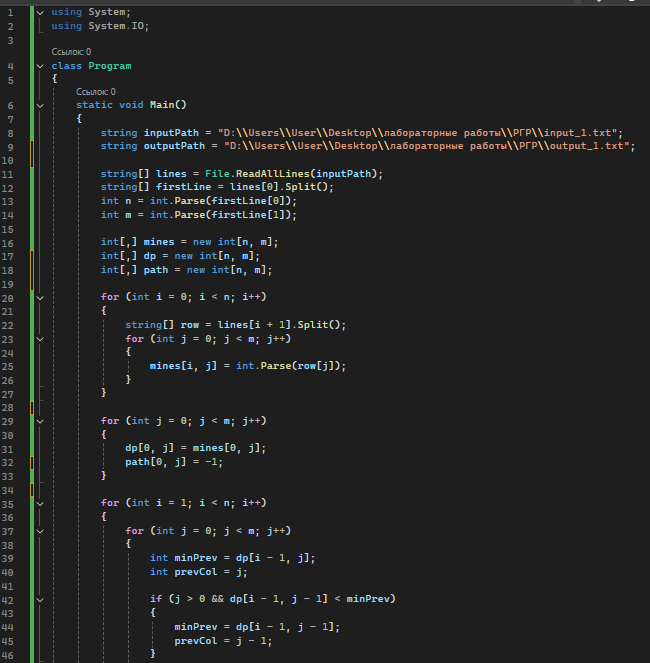
****

Рисунок 9 – Первая часть кода программы «Минное поле»

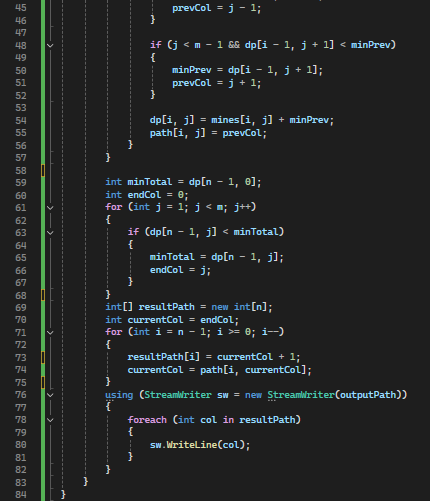
****

Рисунок 10 – Вторая часть кода программы «Минное поле»

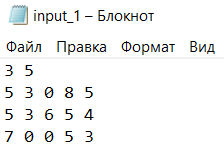
****

Рисунок 11 – Пример ввода программы «Минное поле»

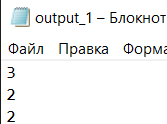
****

Рисунок 12 – Пример вывода программы «Минное поле»

# **Постановка задачи «Расстояние между городами»**

На поверхности некоторой планеты, представляющей собой идеальный шар радиуса R, заданы координаты двух городов в виде двух чисел - широты и долготы.

Требуется определить минимальное расстояние по поверхности планеты между этими городами.

Примечание

Пары (широта, долгота) уникальны.

Входной файл

Первая строка содержит два целых числа S1 и D1 - широту и долготу первого города в градусах (-90  S1  90; 0  D1  359).

Вторая строка содержит два целых числа S2 и D2 - широту и долготу второго города в градусах (-90  S2  90; 0  D2  359).

Третья строка содержит целое число R - радиус планеты (1  R  30000).

Выходной файл

Должен содержать одно вещественное число - минимальное расстояние между городами по поверхности планеты, выведенное с тремя знаками после запятой.

Пример:

Input.txt

-45 0

45 180

1000

Output.txt

3141.593

# **Ход задачи «Расстояние между городами»**

Вычислить расстояние между двумя точками на сфере (например, на планете), зная их широту и долготу в градусах, а также радиус сферы.  
Расчёт ведётся с использованием формулы центрального угла между двумя точками на сфере.

1. Ввод данных: считывание всех строк из входного файла input\_1.txt. Первая строка – координаты первой точки: широта и долгота. Вторая строка – координаты второй точки: широта и долгота. Третья строка – радиус сферы.
2. Проверка формата: убедимся, что широта и долгота заданы правильно.
3. Преобразование в числа: преобразуем строки в целые числа:
4. Проверка диапазонов.
5. Перевод в радианы: широта и долгота преобразуются в радианы, т.к. функции Math.Sin, Math.Cos принимают аргументы в радианах.
6. Вычисление центрального угла
7. Защита от погрешностей: обрезка результата, чтобы он не вышел за допустимый диапазон [-1; 1] из-за ошибок округления.
8. Расчёт расстояния: получаем угол в радианах, затем умножаем на радиус – получаем длину дуги (расстояние).
9. Вывод результата: в файл записывается номер столбца в каждой строке пути.

Код программы «Прямоугольник» представлен на рисунках 13 и 14:

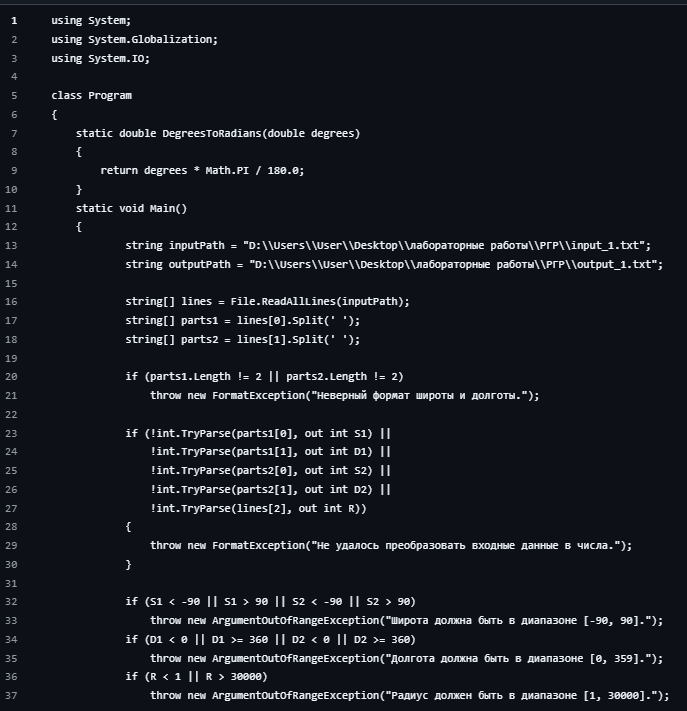


Рисунок 13 – Первая часть кода программы «Расстояние между городами»

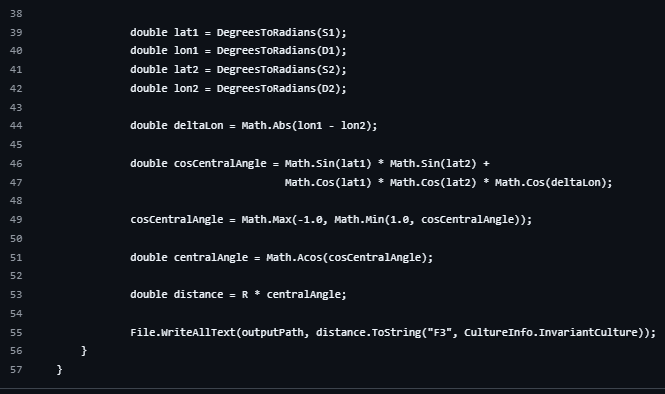


Рисунок 14 – Вторая часть кода программы «Расстояние между городами»

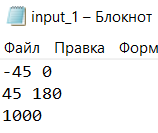


Рисунок 15 – Пример ввода программы «Расстояние между городами»

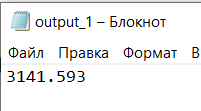


Рисунок 16 – Пример вывода программы «Расстояние между городами»

# **Заключение**

C# – это объектно-ориентированный, строго типизированный язык программирования, разработанный Microsoft, являющийся ключевым для платформы .NET. Он компилируется в промежуточный язык (IL), который затем выполняется .NET CLR, обеспечивая кроссплатформенность на Windows, Linux и macOS. C# широко используется для разработки настольных Windows-приложений, веб-приложений ([ASP.NET](https://asp.net/) Core), игр (Unity), мобильных приложений (.NET MAUI), облачных сервисов (Azure) и в области AI. Одной из важных особенностей C# является автоматическое управление памятью через сборщик мусора, освобождающий разработчиков от ручного управления памятью. C# интегрирует LINQ (Language Integrated Query) для удобной работы с данными, позволяя выполнять запросы непосредственно в коде. Для обеспечения отзывчивости приложений C# поддерживает асинхронное программирование с async и await. Язык имеет простой синтаксис, что делает его сравнительно легким в изучении. Он предоставляет обширную библиотеку классов .NET для решения разнообразных задач. C# обеспечивает высокую производительность благодаря JIT-компиляции. Существует большое и активное сообщество разработчиков, обеспечивающее поддержку и ресурсы. C# хорошо интегрирован с продуктами Microsoft, что является преимуществом для разработки Windows-ориентированных решений. Он подходит для создания как простых, так и сложных проектов. Разнообразие областей применения делает его привлекательным для многих разработчиков. C# продолжает развиваться, становясь еще более актуальным и удобным для работы. Он остается одним из ключевых языков программирования в современной разработке программного обеспечения.

# **Список используемой литературы**

1. Васильев А. Н. Программирование на C# для начинающих. - Москва: Эксмо, 2019. - 525 с.
2. Джон Пол Мюллер, Билл Семпф, Чак Сфер C# для чайников. - СБП: Диалектика, 2019. - 596 с.
3. Стиллимен Эндрю, Грин Дженнифер Head First. Изучаем C#. - СБП: Питер, 2022. - 769 с.
4. C# и .NET // METANIT.COM URL: https://metanit.com/sharp/tutorial/2.7.php (дата обращения: 03.11.2024).
5. Документация по языку C# // Mcrosoft.com URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/ (дата обращения: 03.11.2024).