кдз 7

using System;

public class MazeGenerator

{

private static Random random = new Random();

public static char[,] GenerateMaze(int width, int height)

{

if (width % 2 == 0 || height % 2 == 0)

{

throw new ArgumentException("Width and height must be odd numbers.");

}

char[,] maze = new char[height, width];

for (int y = 0; y < height; y++)

{

for (int x = 0; x < width; x++)

{

maze[y, x] = '#';

}

}

Divide(maze, 0, 0, height - 1, width - 1);

maze[0, 1] = ' ';

maze[height - 1, width - 2] = ' ';

return maze;

}

private static void Divide(char[,] maze, int minX, int minY, int maxX, int maxY)

{

if (maxX - minX < 3 || maxY - minY < 3)

{

return;

}

bool horizontal = random.Next(2) == 0;

int divisionPoint;

if (horizontal)

{

divisionPoint = random.Next(minY + 1, maxY - 1);

for (int x = minX; x <= maxX; x++)

{

maze[divisionPoint, x] = '#';

}

int passageX = random.Next(minX + 1, maxX + 1);

if (passageX % 2 != 0)

{

maze[divisionPoint, passageX] = ' ';

}

Divide(maze, minX, minY, maxX, divisionPoint - 1);

Divide(maze, minX, divisionPoint + 1, maxX, maxY);

}

else

{

divisionPoint = random.Next(minX + 1, maxX - 1);

for (int y = minY; y <= maxY; y++)

{

maze[y, divisionPoint] = '#';

}

int passageY = random.Next(minY + 1, maxY + 1);

if (passageY % 2 != 0)

{

maze[passageY, divisionPoint] = ' ';

}

Divide(maze, minX, minY, divisionPoint - 1, maxY);

Divide(maze, divisionPoint + 1, minY, maxX, maxY);

}

}

public static void PrintMaze(char[,] maze)

{

for (int y = 0; y < maze.GetLength(0); y++)

{

for (int x = 0; x < maze.GetLength(1); x++)

{

Console.Write(maze[y, x]);

}

Console.WriteLine();

}

}

public static void Main(string[] args)

{

int width = 21;

int height = 15;

char[,] maze = GenerateMaze(width, height);

PrintMaze(maze);

}

}

Console.ReadKey();

кдз 9

using System;

using System.Diagnostics;

using System.Linq;

public class MemoryGame

{

private static Random random = new Random();

private static char[,] board;

private static char[,] revealedBoard;

private static bool[,] matched;

private static int boardSize;

private static Stopwatch stopwatch;

public static void Main(string[] args)

{

Console.WriteLine("Введите размер поля (2n, например, 4 для 4x4):");

if (!int.TryParse(Console.ReadLine(), out boardSize) || boardSize % 2 != 0)

{

Console.WriteLine("Некорректный размер поля. Завершение работы.");

return;

}

boardSize /= 2; *// Adjust to half size for pair calculations*

board = GenerateBoard(boardSize);

revealedBoard = new char[boardSize \* 2, boardSize \* 2];

matched = new bool[boardSize \* 2, boardSize \* 2];

InitializeRevealedBoard();

stopwatch = Stopwatch.StartNew();

PlayGame();

}

private static char[,] GenerateBoard(int halfSize)

{

int numPairs = halfSize \* halfSize;

char[] symbols = Enumerable.Range(0, numPairs).Select(i => (char)('A' + i)).ToArray();

char[] shuffledSymbols = symbols.Concat(symbols).OrderBy(x => random.Next()).ToArray();

char[,] board = new char[halfSize \* 2, halfSize \* 2];

int index = 0;

for (int i = 0; i < halfSize \* 2; i++)

{

for (int j = 0; j < halfSize \* 2; j++)

{

board[i, j] = shuffledSymbols[index++];

}

}

return board;

}

private static void InitializeRevealedBoard()

{

for (int i = 0; i < boardSize \* 2; i++)

{

for (int j = 0; j < boardSize \* 2; j++)

{

revealedBoard[i, j] = '#';

}

}

}

private static void PlayGame()

{

int selectedCards = 0;

int firstCardX = -1, firstCardY = -1;

while (true)

{

PrintBoard();

Console.WriteLine("Введите координаты (строка, столбец):");

if (GetCoordinates(out int x, out int y)) {

if (matched[x, y]) {

Console.WriteLine("Эта пара уже найдена!");

continue;

}

revealedBoard[x, y] = board[x, y];

selectedCards++;

if (selectedCards == 1)

{

firstCardX = x;

firstCardY = y;

}

else

{

if (board[x, y] == board[firstCardX, firstCardY])

{

matched[x, y] = true;

matched[firstCardX, firstCardY] = true;

selectedCards = 0;

}

else

{

Console.WriteLine("Не совпадение!");

System.Threading.Thread.Sleep(1000);

revealedBoard[x, y] = '#';

revealedBoard[firstCardX, firstCardY] = '#';

selectedCards = 0;

}

}

if (IsGameOver())

{

break;

}

} else {

Console.WriteLine("Некорректный ввод координат.");

}

}

stopwatch.Stop();

Console.WriteLine($"Поздравляем! Вы выиграли за {stopwatch.ElapsedMilliseconds / 1000.0} секунд!");

}

private static bool GetCoordinates(out int x, out int y) {

string input = Console.ReadLine();

string[] parts = input.Split(',');

if (parts.Length == 2 && int.TryParse(parts[0], out x) && int.TryParse(parts[1], out y) &&

x >= 0 && x < boardSize \* 2 && y >= 0 && y < boardSize \* 2) {

return true;

}

x = -1;

y = -1;

return false;

}

private static bool IsGameOver()

{

for (int i = 0; i < boardSize \* 2; i++)

{

for (int j = 0; j < boardSize \* 2; j++)

{

if (!matched[i, j])

{

return false;

}

}

}

return true;

}

private static void PrintBoard()

{

Console.Clear();

for (int i = 0; i < boardSize \* 2; i++)

{

for (int j = 0; j < boardSize \* 2; j++)

{

Console.Write(revealedBoard[i, j] + " ");

}

Console.WriteLine();

}

}

}

11.1

using System;

using System.IO;

public class SinTableGenerator

{

public static void Main(string[] args)

{

string filePath = "f.txt";

try

{

using (StreamWriter writer = new StreamWriter(filePath))

{

writer.WriteLine("x\tsin(x)");

for (double x = 0; x <= 1; x += 0.1)

{

double sinX = Math.Sin(x);

writer.WriteLine($"{x:F1}\t{sinX:F4}");

}

}

Console.WriteLine($"Таблица синусов успешно записана в файл: {filePath}");

}

catch (Exception ex)

{

Console.WriteLine($"Ошибка при записи в файл: {ex.Message}");

}

}

}

11.1

using System;

using System.Text.RegularExpressions;

public class ThreeZerosChecker

{

public static bool IsStringInSet(string input)

{

string fullMatchRegex = "^.\*000.\*$";

string partialMatchRegex = "000";

bool fullMatch = Regex.IsMatch(input, fullMatchRegex);

bool partialMatch = !fullMatch && Regex.IsMatch(input, partialMatchRegex);

return fullMatch || partialMatch;

}

public static void Main(string[] args)

{

Console.WriteLine("Введите строку:");

string inputString = Console.ReadLine();

if (IsStringInSet(inputString))

{

Console.WriteLine($"Строка '{inputString}' содержит три нуля подряд.");

}

else

{

Console.WriteLine($"Строка '{inputString}' не содержит три нуля подряд.");

}

}

}

4.1

using System;

using System.Collections.Generic;

public class Quiz

{

private List<Question> questions;

private int correctAnswers;

public Quiz(List<Question> questions)

{

this.questions = questions;

this.correctAnswers = 0;

}

public void Start()

{

Console.WriteLine("Добро пожаловать на тест!");

foreach (var question in questions)

{

Console.WriteLine(question.Text);

for (int i = 0; i < question.Options.Length; i++)

{

Console.WriteLine($"{i + 1}. {question.Options[i]}");

}

Console.Write("Введите номер ответа: ");

string answer = Console.ReadLine();

if (int.TryParse(answer, out int choice) && choice >= 1 && choice <= 4 && choice == question.CorrectAnswer)

{

Console.WriteLine("Правильно!\n");

correctAnswers++;

}

else

{

Console.WriteLine($"Неправильно. Правильный ответ: {question.Options[question.CorrectAnswer -1]}\n");

}

}

ShowResults();

}

private void ShowResults()

{

Console.WriteLine($"\nКоличество правильных ответов: {correctAnswers} из {questions.Count}");

double percentage = (double)correctAnswers / questions.Count \* 100;

string result = percentage >= 70 ? "Пройден" : "Не пройден";

Console.WriteLine($"Результат теста: {result} ({percentage:F1}%)");

}

}

public class Question

{

public string Text { get; set; }

public string[] Options { get; set; }

public int CorrectAnswer { get; set; }

public Question(string text, string[] options, int correctAnswer)

{

Text = text;

Options = options;

CorrectAnswer = correctAnswer;

}

}

public class Program

{

public static void Main(string[] args)

{

*// Пример вопросов (замените на свои вопросы)*

List<Question> questions = new List<Question>()

{

new Question("Какой язык программирования мы изучаем?", new string[] { "Java", "Python", "C#", "JavaScript" }, 3),

new Question("Что такое ООП?", new string[] { "Объектно-ориентированное программирование", "Объектно-ориентированный подход", "Ориентированное объектное программирование", "Объектный подход" }, 1),

new Question("Что выводит `Console.WriteLine(\"Hello, world!\");`?", new string[] { "Hello, world!", "Hello world!", "Ничего", "Ошибка" }, 1)

};

Quiz quiz = new Quiz(questions);

quiz.Start();

}

}