***ПУСТОЙ ЛИСТ ДЛЯ ЗАГОТОВОК***

Содержание

[*Задача А 1/1 (Зачтено)* 3](#__RefHeading___Toc7428_1772474102)

[*Задача А 2/1 ИСПРАВЛЕНО (добавлен блок с пояснением)* 5](#__RefHeading___Toc7428_1772474102_Copy_1)

[*Блок с пояснением Задачи А 2/2* 7](#__RefHeading___Toc2673_344495710)

[*Задача А 3/2 (Зачтено)* 8](#__RefHeading___Toc7428_1772474102_Copy_2)

[*Задача А 4/3 (Зачтено)* 10](#__RefHeading___Toc7428_1772474102_Copy_3)

[*Задача А 5/3 ИСПРАВЛЕНО (заменено условие)* 12](#__RefHeading___Toc7428_1772474102_Copy_4)

[*Задача А 6/3 (Зачтено)* 14](#__RefHeading___Toc7428_1772474102_Copy_5)

[*Задача А 7/3 (Зачтено)* 16](#__RefHeading___Toc7428_1772474102_Copy_6)

[*Задача А 8/4 (Зачтено)* 18](#__RefHeading___Toc7428_1772474102_Copy_7)

[*Задача А 9/4* 20](#__RefHeading___Toc7428_1772474102_Copy_8)

[*Задача А 10/5* 22](#__RefHeading___Toc7428_1772474102_Copy_9)

[*Задача А 11/5* 24](#__RefHeading___Toc7428_1772474102_Copy_a)

[*Задача А 12/5* 26](#__RefHeading___Toc7428_1772474102_Copy_b)

# ***Задача А 1/1 (Зачтено)***

***Главная государственная налоговая инспекция страны Х определила следующую шкалу ставок месячного подоходного налога граждан: до 1000 у.е. – не облагается, более 1000-5000 у.е. - 10% суммы дохода, превышающего размер не облагаемого минимума, более 5000 до 10000 – 500 у.е. плюс 20% от суммы, превышающей 5000 у.е., свыше 10000 у.е. – 1000 у.е    плюс 30% от суммы, превышающей 10000 у.е. Требуется определить величину месячного подоходного налога по заданной сумме месячного дохода.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static double CalculateTax(double income)  {  if (income <= 1000) return 0;  if (income <= 5000) return (income - 1000) \* 0.1;  if (income <= 10000) return 500 + (income - 5000) \* 0.2;  return 1000 + (income - 10000) \* 0.3;  }  static void Main()  {  Console.Write("Введите доход: ");  string input = Console.ReadLine() ?? "";  try  {  double income = double.Parse(input);  if (income < 0) throw new ArgumentException("Отрицательный доход");  double tax = CalculateTax(income);  Console.WriteLine($"Налог: {tax} у.е.");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 2/1 (Зачтено)***

***Дан массив вещественных чисел. Выяснить:***

***а) верно ли, что максимальный элемент превышает минимальный не более чем на 25;***

***б) верно ли, что минимальный элемент меньше мак­симального более чем в два раза?***

Решение

|  |  |
| --- | --- |
|  | using System;  class Program  {  static bool CheckDifference(double max, double min) => (max - min) <= 25;  static bool CheckRatio(double max, double min) => min != 0 && max / min > 2;  static void Main()  {  Console.Write("Количество чисел: ");  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) { Console.WriteLine("Положительное число!"); return; }  double min = 0, max = 0;  try  {  for (int i = 0; i < n; i++)  {  Console.Write($"Число {i + 1}: ");  double num = double.Parse(Console.ReadLine() ?? "");  if (i == 0) min = max = num;  else { if (num < min) min = num; if (num > max) max = num; }  }  Console.WriteLine($"Мин: {min}\nМакс: {max}");  Console.WriteLine($"Условие а): {CheckDifference(max, min)}");  Console.WriteLine($"Условие б): {CheckRatio(max, min)}");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

## *Блок с пояснением Задачи А 2/2*

1. Конструкция => в выражении static bool CheckRatio(double max, double min) => min != 0 && max / min > 2;  
   Оператор => в языке C# используется для объявления лямбда-выражений или сокращённой формы методов. В данном случае он применяется для компактной записи логики метода. Метод CheckRatio проверяет, что значение min не равно нулю (во избежание деления на ноль), и что отношение max / min больше 2.
2. Конструкция int n = int.Parse(Console.ReadLine() ?? "");  
   Оператор ?? (null-coalescing operator) используется для обработки возможного значения null, которое может вернуть метод Console.ReadLine(). Если Console.ReadLine() вернёт null, вместо этого будет использована пустая строка "", чтобы избежать ошибки компиляции CS8600: Converting null literal or possible null value to non-nullable type.
3. Модульное тестирование  
   Полноценного модульного тестирования с использованием специализированных библиотек не проводилось, так как разработка велась в Linux-среде (NixOS) без Visual Studio, а фреймворки для модульного тестирования на C# не изучались в рамках курса. Вместо этого была реализована собственная функция RunTests(), которая вручную проверяет корректность работы условий с заранее заданными входными данными и ожидаемыми результатами.
4. Источник изображения с тестами  
   Изображение, демонстрирующее результаты тестирования, было получено с использованием расширения CodeSnap для редактора Visual Studio Code. Сначала запускался тестирующий метод, затем вывод копировался в редактор и из него делался скриншот. Таким образом, иллюстрация создаётся вручную на основе фактического вывода программы.

# ***Задача А 3/2 (Зачтено)***

***Найти все простые несократимые дроби, зак­люченные между 0 и 1, знаменатели которых не превышают 7 (дробь задается двумя натуральными числами - числителем и знаменателем)***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int Gcd(int a, int b)  {  while (b != 0) { int temp = b; b = a % b; a = temp; }  return a;  }  static void Main()  {  Console.WriteLine("Несократимые дроби (0 < x < 1, знаменатель ≤ 7):");  for (int d = 2; d <= 7; d++)  for (int n = 1; n < d; n++)  if (Gcd(n, d) == 1)  Console.WriteLine($"{n}/{d}");  }  } |
|  | |

# ***Задача А 4/3 (Зачтено)***

***Дан массив x [1]..x[n] целых чисел. Не используя других массивов, переставить элементы массива в обратном порядке.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void ReverseArray(int[] arr)  {  for (int i = 0; i < arr.Length / 2; i++)  {  int temp = arr[i];  arr[i] = arr[arr.Length - 1 - i];  arr[arr.Length - 1 - i] = temp;  }  }  static void Main()  {  Console.Write("Введите размер массива: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) { Console.WriteLine("Размер должен быть положительным"); return; }  int[] arr = new int[n];  for (int i = 0; i < n; i++)  {  Console.Write($"Элемент {i + 1}: ");  arr[i] = int.Parse(Console.ReadLine() ?? "");  }  Console.WriteLine("Массив: [" + string.Join(", ", arr) + "]");  ReverseArray(arr);  Console.WriteLine("Обратный массив: [" + string.Join(", ", arr) + "]");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 5/3 (Зачтено)***

***В массиве хранятся значения роста двенадцати человек. С помощью датчика случайных чисел запол­нить массив целыми значениями, лежащими в диапазоне от 163 до 190 включительно.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void FillArrayWithRandomHeights(int[] arr, int min, int max)  {  Random rand = new Random();  for (int i = 0; i < arr.Length; i++)  arr[i] = rand.Next(min, max + 1);  }  static void Main()  {  const int n = 12, minHeight = 163, maxHeight = 190;  int[] heights = new int[n];  FillArrayWithRandomHeights(heights, minHeight, maxHeight);  Console.WriteLine("Рост: [" + string.Join(", ", heights) + "]");  }  } |
|  | |

# ***Задача А 6/3 (Зачтено)***

***В массиве хранятся сведения о количестве осадков, выпавших за каждый день июня. Определить общее количество осадков, выпавших за каждую дека­ду этого месяца.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int SumDecade(int[] arr, int start, int count)  {  int sum = 0;  for (int i = start; i < start + count; i++)  {  if (arr[i] < 0) throw new ArgumentException($"Отрицательное значение в день {i + 1}");  sum += arr[i];  }  return sum;  }  static void Main()  {  int[] rainfall = new int[30];  try  {  for (int i = 0; i < 30; i++)  {  Console.Write($"День {i + 1}: ");  string input = Console.ReadLine() ?? "";  if (!int.TryParse(input, out int value)) throw new FormatException("Введено не число. Пожалуйста, введите целое число.");  rainfall[i] = value;  }  Console.WriteLine($"1-я декада: {SumDecade(rainfall, 0, 10)} мм");  Console.WriteLine($"2-я декада: {SumDecade(rainfall, 10, 10)} мм");  Console.WriteLine($"3-я декада: {SumDecade(rainfall, 20, 10)} мм");  }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 7/3 (Зачтено)***

***Из элементов массива P сформировать мас­сив Q того же размера по правилу: элементы с номе­ром i с 3-го по 10-й находятся по формуле Qi= —Рi,, все остальные — по формуле Qi = Рi\*i.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void FormArrayQ(int[] P, int[] Q)  {  if (P.Length != Q.Length) throw new ArgumentException("Одинаковый размер массивов");  for (int i = 0; i < P.Length; i++)  Q[i] = (i >= 2 && i <= 9) ? -P[i] : P[i] \* i;  }  static void Main()  {  Console.Write("Размер массива: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) throw new ArgumentException("Положительный размер");  int[] P = new int[n], Q = new int[n];  for (int i = 0; i < n; i++)  {  Console.Write($"Элемент {i + 1}: ");  P[i] = int.Parse(Console.ReadLine() ?? "");  }  FormArrayQ(P, Q);  Console.WriteLine("P: [" + string.Join(", ", P) + "]");  Console.WriteLine("Q: [" + string.Join(", ", Q) + "]");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 8/4 (Зачтено)***

***Дана квадратная таблица a[1:n,1:n]. Переставить местами строки 1 и 2, 3 и 4 и т.д. Если n нечетно, то последняя строка остается на месте.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void SwapRows(int[,] a, int n)  {  if (n <= 0 || a.GetLength(0) != n || a.GetLength(1) != n)  throw new ArgumentException("Квадратная таблица с n > 0");  for (int i = 0; i < n - 1; i += 2)  for (int j = 0; j < n; j++)  {  int temp = a[i, j];  a[i, j] = a[i + 1, j];  a[i + 1, j] = temp;  }  }  static void Main()  {  Console.Write("Размер таблицы: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  if (n <= 0) throw new ArgumentException("Положительный размер");  int[,] a = new int[n, n];  for (int i = 0; i < n; i++)  for (int j = 0; j < n; j++)  {  Console.Write($"Элемент [{i + 1},{j + 1}]: ");  a[i, j] = int.Parse(Console.ReadLine() ?? "");  }  Console.WriteLine("Исходная:");  for (int i = 0; i < n; i++)  {  int[] row = new int[n];  for (int j = 0; j < n; j++) row[j] = a[i, j];  Console.WriteLine("[" + string.Join(", ", row) + "]");  }  SwapRows(a, n);  Console.WriteLine("После обмена:");  for (int i = 0; i < n; i++)  {  int[] row = new int[n];  for (int j = 0; j < n; j++) row[j] = a[i, j];  Console.WriteLine("[" + string.Join(", ", row) + "]");  }  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 9/4 (Зачтено)***

***Дан двумерный массив размерностью 5 х 6, заполненный целыми числами с клавиатуры. Сформировать одномерный массив, каждый элемент которого равен наибольшему по модулю элементу соответствующего столбца.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void FormMaxAbsArray(int[,] a, int[] result)  {  if (a.GetLength(0) != 5 || a.GetLength(1) != 6 || result.Length != 6)  throw new ArgumentException("Размер матрицы 5x6, размер массива результат — 6");  for (int j = 0; j < 6; j++)  {  int maxAbs = Math.Abs(a[0, j]);  for (int i = 1; i < 5; i++)  if (Math.Abs(a[i, j]) > maxAbs)  maxAbs = Math.Abs(a[i, j]);  result[j] = maxAbs;  }  }  static void Main()  {  int[,] a = new int[5, 6];  int[] result = new int[6];  try  {  for (int i = 0; i < 5; i++)  for (int j = 0; j < 6; j++)  {  Console.Write($"Элемент [{i + 1},{j + 1}]: ");  a[i, j] = int.Parse(Console.ReadLine() ?? "");  }  FormMaxAbsArray(a, result);  Console.WriteLine("Матрица: ");  for (int i = 0; i < 5; i++)  {  int[] row = new int[6];  for (int j = 0; j < 6; j++) row[j] = a[i, j];  Console.WriteLine("[" + string.Join(", ", row) + "]");  }  Console.WriteLine("Результат: [" + string.Join(", ", result) + "]");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 10/5 (Зачтено)***

***Дано слово. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, то есть в слове должны остаться только различные буквы.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static string RemoveDuplicates(string word)  {  if (string.IsNullOrEmpty(word)) throw new ArgumentException("Пустое слово");  string result = "";  for (int i = 0; i < word.Length; i++)  {  bool isDuplicate = false;  for (int j = 0; j < result.Length; j++)  {  if (word[i] == result[j])  {  isDuplicate = true;  break;  }  }  if (!isDuplicate)  result += word[i];  }  return result;  }  static void Main()  {  Console.Write("Введите слово: ");  try  {  string input = Console.ReadLine() ?? "";  string result = RemoveDuplicates(input);  Console.WriteLine($"Слово: {input}\nРезультат: {result}");  }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 11/5 (Зачтено)***

***Составить программу подсчета общего количества цифр и знаков '+’, '-‘, '\*' в строке s, введенной с клавиатуры.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int CountDigitsAndSigns(string s)  {  if (string.IsNullOrEmpty(s)) throw new ArgumentException("Пустая строка");  int count = 0;  for (int i = 0; i < s.Length; i++)  {  char c = s[i];  if (char.IsDigit(c) || c == '+' || c == '-' || c == '\*')  count++;  }  return count;  }  static void Main()  {  Console.Write("Введите строку: ");  try  {  string input = Console.ReadLine() ?? "";  int result = CountDigitsAndSigns(input);  Console.WriteLine($"Строка: {input}\nРезультат: {result}");  }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 12/5***

***Дано предложение. Все пробелы в нем заменить на символ «\_» .***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static string ReplaceSpaces(string sentence)  {  if (string.IsNullOrEmpty(sentence)) throw new ArgumentException("Пустая строка");  string result = "";  for (int i = 0; i < sentence.Length; i++)  {  char c = sentence[i];  result += c == ' ' ? '\_' : c;  }  return result;  }  static void Main()  {  Console.Write("Введите предложение: ");  try  {  string input = Console.ReadLine() ?? "";  string result = ReplaceSpaces(input);  Console.WriteLine($"Строка: {input}\nРезультат: {result}");  }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача А 13/7 (7 раздел отсутствует)***

***Составить программу, печатающую квадраты всех натуральных чисел от 0 до заданного натурального n, но разрешается использовать из арифметических операций лишь сложение и вычитание, причем общее число действий должно быть порядка n.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static void PrintSquares(int n)  {  if (n < 0) throw new ArgumentException("Неотрицательное число");  int square = 0, diff = 1;  for (int i = 0; i <= n; i++)  {  Console.WriteLine($"{i}² = {square}");  diff += 2;  square += diff - 2;  }  }  static void Main()  {  Console.Write("Введите n: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  PrintSquares(n);  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача B 1/1***

***Найти количество делителей натурального числа. Сколько из них четных.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static (int total, int even) CountDivisors(int n)  {  if (n <= 0) throw new ArgumentException("Натуральное число");  int total = 0, even = 0;  for (int i = 1; i <= n; i++)  if (n % i == 0)  {  total++;  if (i % 2 == 0) even++;  }  return (total, even);  }  static void Main()  {  Console.Write("Введите число: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  var (total, even) = CountDivisors(n);  Console.WriteLine($"Число: {n}\nДелители: {total}, Чётные: {even}");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача B 2/1***

***Функция f с натуральными аргументами и значениями определена так: f(0) = 0, f(1) = 1, …, f (2n) = f(n), f (2n+1) = f (n) + f (n+1). Составить программу вычисления f (n) по заданному n.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static int ComputeF(int n)  {  if (n < 0) throw new ArgumentException("Неотрицательное число");  if (n == 0) return 0;  int coeff\_a = 1, coeff\_b = 0;  while (n > 0)  {  if (n % 2 == 0)  {  coeff\_a += coeff\_b;  n /= 2;  }  else  {  coeff\_b += coeff\_a;  n = (n - 1) / 2;  }  }  return coeff\_b;  }  static void Main()  {  Console.Write("Введите n: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  int result = ComputeF(n);  Console.WriteLine($"Число: {n}\nРезультат: {result}");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |

# ***Задача B 3/2***

***Вычислить произведение ряда*** ***где n – натуральное число.***

Решение:

|  |  |
| --- | --- |
|  | using System;  class Program  {  static double ComputeProduct(int n)  {  if (n <= 0) throw new ArgumentException("Натуральное число");  double product = 1.0;  for (int i = 1; i <= n; i++)  {  double factorial = 1.0;  for (int j = 1; j <= i; j++) factorial \*= j;  product \*= 2.0 + 1.0 / factorial;  }  return product;  }  static void Main()  {  Console.Write("Введите n: ");  try  {  int n = int.Parse(Console.ReadLine() ?? "");  double result = ComputeProduct(n);  Console.WriteLine($"Число: {n}\nРезультат: {result:F10}");  }  catch (FormatException) { Console.WriteLine("Ошибка: Некорректное число"); }  catch (ArgumentException ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  catch (Exception ex) { Console.WriteLine($"Ошибка: {ex.Message}"); }  }  } |
|  | |