Дисциплина «Программирование корпоративных систем» Рабочая тетрадь 2

Циклы

|  |
| --- |
| **Теоретический материал** |
| **Циклы:**  Циклы являются управляющими конструкциями, позволяя в зависимости от определенных условий выполнять некоторое действие множество раз. В C# имеются следующие виды циклов:   * **for** * **foreach** * **while** * **do...while**   **Цикл for**  for([действия\_до\_выпол-ния\_цикла];[условие];[действия\_после\_выпол-ния])  {  // действия  }  **Цикл do..while**  В цикле **do** сначала выполняется код цикла, а потом происходит проверка условия в инструкции **while**. И пока это условие истинно, цикл повторяется.  do  {  //действия цикла  }  while (условие);  **Цикл while**  Цикл **while** сразу проверяет истинность некоторого условия, и если условие истинно, то код цикла выполняется.  while (условие)  {       //действия цикла  }  **Цикл foreach**  Цикл **foreach** предназначен для перебора набора или коллекции элементов. Его общее определение:  foreach(тип\_данных переменная in коллекция)  {       // действия цикла  }  **Операторы break и continue**  Иногда возникает ситуация, когда требуется выйти из цикла, не дожидаясь его завершения. В этом случае мы можем воспользоваться оператором **break**.  for (int i = 0; i < 9; i++)  {  if (i == 5)  break;  Console.WriteLine(i);  }  Если необходимо, чтобы при проверке цикл не завершался, а просто пропускал текущую итерацию. Для этого можно воспользоваться оператором **continue**:  for (int i = 0; i < 9; i++)  {  if (i == 5)  continue;  Console.WriteLine(i);  }  Одни циклы могут быть вложенными в другие:  for (int i = 1; i < 10; i++)  {  for (int j = 1; j < 10; j++)  {  Console.Write($"{i \* j} \t");  }  Console.WriteLine();  } |

|  |  |
| --- | --- |
|  | |
| **Задание 1** | |
| ***Задача:*** | |
|  | **Ряды**  Реализуйте программный продукт средствами языка C# со следующим функционалом:  Вычисление значения функции f(x) (соответствующей вашему варианту) с помощью ряда Маклорена с заданной точностью е (e и x вводятся с клавиатуры, е <0.01);  Вычисление n-го члена ряда (n и x вводятся с клавиатуры). |
| ***Решение:*** | |
|  | using System;  Console.Write("Введите x для вычисления sin(x) рядом Маклорена: ");  double x1 = double.Parse(Console.ReadLine());  Console.Write("Введите точность e (e < 0,01, иначе значение по умолчанию 0,01): ");  double e = double.Parse(Console.ReadLine());  if (e > 0.01) e = 0.01;  double result = CalculateSin(x1, e);  Console.WriteLine($"sin({x1}) ≈ {result}");  Console.Write("Введите x для вычисления n-го члена ряда: ");  double x2 = double.Parse(Console.ReadLine());  Console.Write("Введите n для вычисления n-го члена ряда: ");  int n = int.Parse(Console.ReadLine());  double nthTerm = CalculateNthTerm(x2, n);  Console.WriteLine($"{n}-й член ряда: {nthTerm}");  double CalculateSin(double x, double e)  {  double term = x;  double sum = term;  int n = 1;  term = Math.Pow(-1, n) \* (Math.Pow(x, 2 \* n + 1) / Factorial(2 \* n + 1));  while (Math.Abs(term) > e)  {  n++;  sum += term;  term = Math.Pow(-1, n) \* (Math.Pow(x, 2 \* n + 1) / Factorial(2 \* n + 1));  }  return sum;  }  double CalculateNthTerm(double x, int n)  {  return Math.Pow(-1, n) \* Math.Pow(x, 2 \* n + 1) / Factorial(2 \* n + 1);  }  double Factorial(int n)  {  if (n == 0 || n == 1)  return 1;  return n \* Factorial(n - 1);  } |
| ***Ответ:*** | |
|  |  |

|  |  |
| --- | --- |
| **Задание 2** | |
| ***Задача:*** | |
|  | **Счастливый билет**  Если на билете сумма первых трёх цифр в номере билета равна сумме трёх последних, то этот билет считается счастливым. Напишите программу, которая получала бы на вход шестизначный номер билета и выводила, счастливый это билет или нет. К примеру: билеты 777 777 и 255 642 — счастливые, а 123 456 — нет.  Использовать при решении задачи можно только простые базовые типы (т.е. использование массивов, строк и коллекций запрещено, должно обрабатываться именно число).  ***Пример выполнения:***  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите номер билета: 123060*  *True*  *Введите номер билета: 123040*  *False*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  Совет: Получайте разряды числа за счет использования операций «/» и «%». |
| ***Решение:*** | |
|  | using System;  while (true)  {  Console.Write("Введите номер билета: ");  int bilet = int.Parse(Console.ReadLine());  int first\_half = 0;  int second\_half = 0;  int count\_of\_numbers = 6;  for (int i = 0; i < count\_of\_numbers; i++)  {  if (i < count\_of\_numbers / 2)  {  second\_half += (bilet % 10);  bilet /= 10;  }  else  {  first\_half += bilet % 10;  bilet /= 10;  }  }  if (first\_half == second\_half) Console.WriteLine("True");  else Console.WriteLine("False");  }  } |
| ***Ответ:*** | |
|  |  |
| **Задание 3** | |
| ***Задача:*** | |
|  | **Сокращение дроби**  Пользователь вводит числа M и N. Напишите программу, которая преобразует дробь M/N к несократимому виду и выдаёт получившийся результат.  ***Пример выполнения:***  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите числитель: 4*  *Введите знаменатель: 6*  *Результат: 2 / 3*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите числитель: 25*  *Введите знаменатель: 40*  *Результат: 5 / 8*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите числитель: −6*  *Введите знаменатель: 12*  *Результат: −1 / 2*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите числитель: 896*  *Введите знаменатель: 3584*  *Результат: 1 / 4*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  (При вводе отрицательных чисел в выводе должен быть только один ведущий минус. Неправильный вывод программы: 1/−2, −1/−2.).  Совет: знаменатель не может быть равен 0. |
| ***Решение:*** | |
|  | using System;  while (true)  {  Console.Write("Введите числитель: ");  int numerator = int.Parse(Console.ReadLine());  Console.Write("Введите знаменатель: ");  int denominator = int.Parse(Console.ReadLine());  if (denominator == 0)  {  Console.WriteLine("Ошибка! Знаменатель не может быть равен 0.");  continue;  }  for (int i = Math.Max(Math.Abs(numerator), Math.Abs(denominator)); i > 1; i--)  {  if (numerator % i == 0 && denominator % i == 0)  {  numerator /= i;  denominator /= i;  }  }  if ((numerator > 0 & denominator > 0) || (numerator < 0 & denominator < 0))  {  Console.WriteLine($"Результат: {Math.Abs(numerator)} / {Math.Abs(denominator)}");  }  else  {  Console.WriteLine($"Результат: - {Math.Abs(numerator)} / {Math.Abs(denominator)}");  }  } |
| ***Ответ:*** | |
|  |  |
| **Задание 4** | |
| ***Задача:*** | |
|  | **Угадай число**  Напишите программу, которая угадывает число, задуманное пользователем. Число загадывается в диапазоне от 0 до 63. Программа задаёт вопросы вида «Ваше число больше такого-то?» и на основе ответов пользователя («да-1» или «нет-0») угадывает число.  Алгоритм, должен давать ответ за семь вопросов. |
| ***Решение:*** | |
|  | using System;  Console.WriteLine("Загадайте число от 0 до 63.");  Console.WriteLine("Отвечайте '1' для 'да' и '0' для 'нет'.");  int min = 0;  int max = 63;  int guessCount = 0;  while (min <= max)  {  int guess = (min + max) / 2;  guessCount++;  Console.Write($"Вопрос {guessCount}: Ваше число больше {guess}? ");  string answer = Console.ReadLine();  if (answer == "1")  {  min = guess + 1;  }  else if (answer == "0")  {  if (min == max)  {  Console.WriteLine($"Ваше число - {guess}!");  break;  }  max = guess;  }  else  {  Console.WriteLine("Пожалуйста, введите '1' для 'да' или '0' для 'нет'.");  guessCount--;  }  } |
| ***Ответ:*** | |
|  |  |
| **Задание 5** | |
| ***Задача:*** | |
|  | **Кофейный аппарат**  Кофейный аппарат может готовить два напитка: американо и латте. Для американо требуется 300 мл воды (цена 150 рублей), а для латте 30 мл воды и 270 мл молока (цена 170 рублей).  Напишите программу, которая спрашивает у пользователя (это действие программа делает один раз в начале работы), сколько всего миллилитров молока и воды залито в кофейный аппарат.  После чего начинает обслуживание пользователей, запрашивается, какой напиток хочет заказать посетитель. Пользователь выбирает один из двух напитков, программа отвечает одним из трёх вариантов: «Ваш напиток готов», «Не хватает воды» или «Не хватает молока», после чего переходит к обслуживанию следующего посетителя. Если молока и воды не хватает ни на один вид напитка, программа выдаёт отчёт и завершается.  В отчёте должно быть написано, что ингредиенты подошли к концу, должен быть указан остаток воды и молока в машине, должно быть указано, сколько всего было приготовлено чашек американо и латте за эту смену и итоговый заработок аппарата.  ***Пример выполнения:***  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите количество воды в мл: 30*  *Введите количество молока в мл: 270*  *Выберите напиток (1 — американо, 2 — латте): 2*  *Ваш напиток готов.*  *\*Отчёт\**  *Ингредиентов осталось:*  *Вода: 0 мл*  *Молоко: 0 мл*  *Кружек американо приготовлено: 0*  *Кружек латте приготовлено: 1*  *Итого: 170 рублей.*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**** |
| ***Решение:*** | |
|  | using System;  int total\_amount = 0;  int total\_americano = 0;  int total\_latte = 0;  Console.Write("Введите количество воды в мл: ");  int total\_water = int.Parse(Console.ReadLine());  Console.Write("Введите количество молока в мл: ");  int total\_milk = int.Parse(Console.ReadLine());  while ((total\_water >= 30 & total\_milk >= 270) || (total\_water >= 300))  {  Console.Write("Выберите напиток (1 - американо, 2 - латте): ");  int user\_choice = int.Parse(Console.ReadLine());  switch (user\_choice)  {  case 1:  if (total\_water >= 300)  {  total\_amount += 150;  total\_water -= 300;  total\_americano += 1;  Console.WriteLine("Ваш напиток готов");  }  else  {  Console.WriteLine("Не хватает воды");  }  continue;    case 2:  if (total\_water >= 30 & total\_milk >= 270)  {  total\_amount += 170;  total\_water -= 30;  total\_milk -= 270;  total\_latte += 1;  Console.WriteLine("Ваш напиток готов");  }  else  {  Console.WriteLine("Не хватает молока");  }  continue;    default:  Console.WriteLine("Неправильный выбор, попробуйте снова.");  continue;  }  }  Console.WriteLine("\*Отчёт\*");  Console.WriteLine("Ингридиентов осталось:");  Console.WriteLine($" Вода: {total\_water} мл");  Console.WriteLine($" Молоко: {total\_milk} мл");  Console.WriteLine($"Кружек американо приготовлено: {total\_americano}");  Console.WriteLine($"Кружек латте приготовлено: {total\_latte}");  Console.WriteLine($"Итого: {total\_amount} рублей."); |
| ***Ответ:*** | |
|  |  |
| **Задание 6** | |
| ***Задача:*** | |
|  | **Лабораторный опыт**  В чашку Петри кладут N бактерий и добавляют X капель антибиотика (N и X вводятся с клавиатуры).  Известно, что число бактерий в чашке Петри увеличивается в два раза каждый час, а каждая капля антибиотика в первый час убивает 10 бактерий, во второй час — 9 бактерий, в следующий — 8 и так далее, пока антибиотик не перестанет действовать. Заметьте, что сначала число бактерий увеличивается, а затем только действует антибиотик.  Пользователь вашей программы вводит N и X, а программа печатает на экране, сколько бактерий останется в чашке Петри в конце каждого часа, до тех пор, пока не закончатся бактерии или антибиотик не перестанет действовать.  Цикл не должен быть бесконечным (после того как количество антибиотики или бактерий становиться равным нулю выполнение программы должно быть завершено).  ***Пример выполнения:***  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите количество бактерий: 12*  *Введите количество антибиотика: 1*  *После 1 часа бактерий осталось 14*  *После 2 часа бактерий осталось 19*  *После 3 часа бактерий осталось 30*  *После 4 часа бактерий осталось 53*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**** |
| ***Решение:*** | |
|  | using System;  Console.Write("Введите количество бактерий: ");  int total\_bakteriy = int.Parse(Console.ReadLine());  Console.Write("Введите количество антибиотика: ");  int total\_antibiotic = int.Parse(Console.ReadLine());  int antibiotic\_duration = 10;  while (total\_bakteriy > 0 & antibiotic\_duration > 0)  {  total\_bakteriy \*= 2;  total\_bakteriy -= total\_antibiotic \* antibiotic\_duration;  antibiotic\_duration -= 1;  if (total\_bakteriy > 0) Console.WriteLine($"После {10 - antibiotic\_duration} часа бактерий осталось: {total\_bakteriy}");  else Console.WriteLine($"После {10 - antibiotic\_duration} часа бактерии закончились");  } |
| ***Ответ:*** | |
|  |  |
| **Задание 7** | |
| ***Задача:*** | |
|  | **Колонизация Марса**  Для освоения Марса требуется построить исследовательскую базу. База должна состоять из n одинаковых модулей, каждый из которых представляет собой прямоугольник.  Каждый модуль представляет собой жилой отсек, который имеет форму прямоугольника размером a × b метров. Для повышения надежности модулей инженеры должны добавить вокруг каждого модуля слой дополнительной защиты. Толщина этого слоя должна составлять целое число метров, и все модули должны иметь одинаковую толщину дополнительной защиты.  Модуль с защитой, толщина которой равна d метрам, будет иметь форму прямоугольника размером (a + 2d) × (b + 2d) метров.  Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером w × h метров. При этом они должны быть организованы в виде регулярной сетки: их стороны должны быть параллельны сторонам поля, и модули должны быть ориентированы одинаково.  Требуется написать программу, которая по заданным количеству n и размеру модулей a и b, а также размеру поля h и w для их размещения, определяет максимальную толщину слоя дополнительной защиты d, который можно добавить к каждому модулю. (Все данные вводятся с клавиатуры).  ***Пример выполнения****:*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  *Введите n: 11*  *Введите a: 2*  *Введите b: 2*  *Введите w: 21*  *Введите h: 25*  *Ответ d = 2*  ***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****  Для значений n = 11, a = 2, b = 3, w = 21, h = 25, возможный вариант реализации задачи представлен ниже.  https://it.kgsu.ru/Algorithms/images/im00901.jpg |
| ***Решение:*** | |
|  | using System;  Console.Write("Введите n: ");  int n = int.Parse(Console.ReadLine());  Console.Write("Введите a: ");  int a = int.Parse(Console.ReadLine());  Console.Write("Введите b: ");  int b = int.Parse(Console.ReadLine());  Console.Write("Введите w: ");  int w = int.Parse(Console.ReadLine());  Console.Write("Введите h: ");  int h = int.Parse(Console.ReadLine());  int left = 0;  int right = Math.Min(w, h) / 2;  int d = 0;  while (left <= right)  {  int mid = (left + right) / 2;  int moduleWidth = a + 2 \* mid;  int moduleHeight = b + 2 \* mid;  int modulesPerRow = w / moduleWidth;  int modulesPerColumn = h / moduleHeight;  if (modulesPerRow \* modulesPerColumn >= n)  {  d = mid;  left = mid + 1;  }  else  {  right = mid - 1;  }  }  Console.WriteLine($"Ответ d = {d}"); |
| ***Ответ:*** | |
|  |  |