# 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;  class Program  {  static double i\_dont\_know\_how\_to\_call\_you(ref double x, ref int n)  {  double temp = 1;  for (int i = 1; i <= 2 \* n + 1; ++i)  {  temp \*= x / i;  }  return temp;  }  static double Sihn(ref double x, ref double eps)  {  double sum = 0, temp = x;  int iter = 0;  while(Math.Abs(temp) > eps)  {  sum += temp;  ++iter;  temp \*= x \* x / ((2 \* iter) \* (2 \* iter + 1));  }  return sum;  }  static void Main()  {  Console.WriteLine("Введите x");  double x = double.Parse(Console.ReadLine() ?? "");  Console.Write("Введите eps (eps < 0.01)");  double eps = double.Parse(Console.ReadLine() ?? "");  double sin = Sihn(ref x, ref eps);  Console.WriteLine($"sinh({x}) = {sin}");  Console.WriteLine("Введите N: ");  int n = int.Parse(Console.ReadLine() ?? "") -1;  double temp\_n = i\_dont\_know\_how\_to\_call\_you(ref x, ref n);  Console.WriteLine($"{n+1}-й член ряда = {temp\_n}");  }  } |
| ***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;  class Program  {  static int sum\_of\_numbers(ref int number)  {  int counter = 0;  while (number != 0)  {  counter += number % 10;  number /= 10;  }  return counter;  }  static void Main()  {  Console.Write("Введите номер билета: ");  int part\_one = int.Parse(Console.ReadLine() ?? "");  int part\_two = part\_one % 1000;  part\_one /= 1000;  Console.WriteLine(sum\_of\_numbers(ref part\_one) == sum\_of\_numbers(ref part\_two) ? "True" : "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;  class Program  {  static int GCD(int a, int b)  {  return b > 0 ? GCD(b, a % b) : a;  }  static void Main()  {  Console.Write("Введите делимое: ");  int dilimiter = int.Parse(Console.ReadLine() ?? "");  Console.Write("Введите делитель: ");  int divisor = int.Parse(Console.ReadLine() ?? "");  int gcd = GCD(Math.Abs(dilimiter), Math.Abs(divisor));  int part\_one = dilimiter / gcd;  int part\_two = divisor / gcd;  part\_one = part\_one == 0 ? 1 : part\_one;  part\_two = part\_two == 0 ? 1 : part\_two;  if(part\_one < 0 && part\_two < 0)  {  part\_one \*= -1;  part\_two \*= -1;  }  else if (part\_two < 0 && part\_one > 0)  {  part\_one \*= -1;  part\_two \*= -1;  }  Console.WriteLine($"{part\_one} / {part\_two}");  }  } |
| ***Ответ:*** | |
|  |  |
| **Задание 4** | |
| ***Задача:*** | |
|  | **Угадай число**  Напишите программу, которая угадывает число, задуманное пользователем. Число загадывается в диапазоне от 0 до 63. Программа задаёт вопросы вида «Ваше число больше такого-то?» и на основе ответов пользователя («да-1» или «нет-0») угадывает число.  Алгоритм, должен давать ответ за семь вопросов. |
| ***Решение:*** | |
|  | using System;  class Program  {  static int bin\_search(ref int left, ref int right)  {  return left + (right - left) / 2;  }  static void Main()  {  int left = -1;  int right = 64;  char ans = '0';  int temp\_bin = 0;  while(left < right - 1)  {  temp\_bin = bin\_search(ref left, ref right);  Console.Write($"Ваше число больше чем {temp\_bin}? ");  ans = char.Parse(Console.ReadLine() ?? "");  switch (ans)  {  case '1':  left = temp\_bin;  break;  case '0':  right = temp\_bin;  break;  default:  break;  }  }  Console.WriteLine($"Ваше число: {right}");  }  } |
| ***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;  class Program  {  static void Main()  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  int counter\_americano = 0; // 150 руб, 300 воды  int counter\_latte = 0; // 170 руб, 30 воды, 270 молока.  Console.Write("\tВведите количество воды: ");  int quantity\_of\_water = int.Parse(Console.ReadLine() ?? "");  Console.Write("\tВведите количество молока: ");  int quantity\_of\_milk = int.Parse(Console.ReadLine() ?? "");  char choise = '0';  while(quantity\_of\_water >= 30 || quantity\_of\_milk >= 270)  {  Console.Write("\tВыберете напиток (1 - американо, 2 - латте): ");  choise = char.Parse(Console.ReadLine() ?? "");  switch (choise)  {  case '1':  if (quantity\_of\_water < 300)  {  Console.WriteLine("\tНе хватает воды!");  }  else  {  quantity\_of\_water -= 300;  ++counter\_americano;  Console.WriteLine("\tВаш напиток готов! ");  }  break;  case '2':  if (quantity\_of\_milk < 270)  Console.WriteLine("\tНе хватает молока!");  else  {  ++counter\_latte;  quantity\_of\_milk -= 270;  quantity\_of\_water -= 30;  Console.WriteLine("\tВаш напиток готов!");  }  break;  }  }  Console.WriteLine("\t\*Отчёт\*");  Console.WriteLine($"\tИнгридинтов осталось\n\t\tВода: {quantity\_of\_water} мл\n\t\tМолоко: {quantity\_of\_milk} мл");  Console.WriteLine($"\tКружек американо приготовлено: {counter\_americano}");  Console.WriteLine($"\tКружек латте приготовлено: {counter\_latte}");  Console.WriteLine($"Итого: {counter\_latte \* 170 + counter\_americano \* 150} рублей.");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  }  } |
| ***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;  class Program  {  static void Main()  {  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  Console.Write("\tВведите количество бактерий: ");  int N = int.Parse(Console.ReadLine() ?? "");  Console.Write("\tВведите количество капель антибиотика: ");  int X = int.Parse(Console.ReadLine() ?? "");  for(int iter = 10; iter >0; iter--)  {  N <<= 1;  N -= X \* iter;  if (N <= 0)  {  Console.WriteLine("\tВсе бедные бактерии сдохли!!!");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  return;  }  else  {  Console.WriteLine($"\tПосле {10 - iter + 1} часа бактерий осталось {N}");  }  }  Console.WriteLine("\tУра бактерии победили!!!!!!!!!!!!!!!!!!!!");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  }  } |
| ***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;  class Program  {  static void cin(ref int value, string promt = "Число")  {  Console.Write($"\tВведите {promt}: ");  value = int.Parse(Console.ReadLine() ?? "");  }  static void Main()  {  int n, a, b, w, h;  n = a = b = w = h = 0;  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  cin(ref n, "n");  cin(ref a, "a");  cin(ref b, "b");  cin(ref w, "w");  cin(ref h, "h");  int left = 0;  int right = Math.Min(w, h);  int res = 0;  while(left <= right)  {  int mid = left + (right - left) / 2;  int temp\_A = 2 \* mid + a;  int temp\_B = 2 \* mid + b;  int counter\_A = w / temp\_A;  int counter\_B = h / temp\_B;  if (counter\_A \* counter\_B >= n)  {  res = mid;  left = mid + 1;  }  else right = mid - 1;  }  Console.WriteLine($"\tОтвет: {res}");  Console.WriteLine("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  }  } |
| ***Ответ:*** | |
|  |  |