

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_выполнения_цикла];[условие];[действия_после_выполнения])  
{  
    // 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.WriteLine($"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 ϵ (ϵ и x вводятся с клавиатуры, $\epsilon < 0.01$);

Вычисление n -го члена ряда (n и x вводятся с клавиатуры).

Вариант №	Ряд
1, 11, 21, 31.	$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots, \forall x \in (-\infty; +\infty)$
2, 12, 22, 32.	$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n}, \forall x \in (-1; 1]$
3, 13, 23, 33.	$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots, \forall x \in (-\infty; +\infty)$
4, 14, 24, 34.	$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots, \forall x \in (-\infty; +\infty)$
5, 15, 25, 35.	$\arcsin x = x + \frac{x^3}{2!! \cdot 3} + \frac{3!! x^5}{4!! \cdot 5} + \frac{5!! x^7}{6!! \cdot 7} + \dots + \frac{(2n-1)!! x^{2n+1}}{(2n)!! (2n+1)} + \dots, \forall x \in (-1; 1)$
6, 16, 26, 36.	$\operatorname{arctg} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots, \forall x \in (-1; 1)$
7, 17, 27, 37.	$\operatorname{sh} x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots, \forall x \in (-\infty; +\infty)$
8, 18, 28, 38.	$\operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2n}}{(2n)!} + \dots, \forall x \in (-\infty; +\infty)$

	9, 19, 29, 39.	$\arsh x = x - \frac{x^3}{2!! \cdot 3} + \frac{3!!x^5}{4!! \cdot 5} - \frac{5!!x^7}{6!! \cdot 7} + \dots + (-1)^n \frac{(2n-1)!!x^{2n+1}}{(2n)!!(2n+1)} + \dots, \forall x \in (-1; 1)$
	10, 20, 30, 40.	$\operatorname{arcth} x = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots + \frac{x^{2n+1}}{2n+1} + \dots, \forall x \in (-1; 1)$

Решение:

```

using System;

namespace Task1
{
    class Program
    {
        static void Main()
        {
            Console.OutputEncoding = System.Text.Encoding.UTF8;
            Console.WriteLine("\r\n\tРеализуйте программный продукт средствами языка C# со следующим
функционалом:\r\n\tВычисление значения функции f(x) (соответствующей вашему варианту) с помощью
ряда Маклорена с заданной точностью ε (ε и x вводятся с клавиатуры, ε < 0.01);\r\n\tВычисление n-го члена
ряда (n и x вводятся с клавиатуры).");
            Console.WriteLine("Программа вычисления e^x с помощью ряда Маклорена");

            while (true)
            {
                Console.WriteLine("\nВыберите действие:");
                Console.WriteLine("1. Вычислить e^x с заданной точностью ε");
                Console.WriteLine("2. Вычислить n-й член ряда");
                Console.WriteLine("3. Выйти");
                Console.Write("Ваш выбор: ");
                string? choice = Console.ReadLine();
                switch (choice)
                {
                    case "1":
                        CalculateWithPrecision();
                        break;
                    case "2":
                        CalculateNthTerm();
                        break;
                    case "3":
                        Console.WriteLine("Выход из программы.");
                        return;
                    default:
                        Console.WriteLine("Неверный выбор. Попробуйте снова.");
                        break;
                }
            }
        }

        static void CalculateWithPrecision()
        {
            double x = ReadDouble("Введите x: ");
            double eps = ReadDouble("Введите точность ε (0 < ε < 0,01): ");

            if (eps <= 0 || eps >= 0.01)
            {
                Console.WriteLine("Ошибка: ε должно быть в диапазоне (0, 0,01)");
                return;
            }

            double sum = 1.0;
            double element = 1.0;
        }
    }
}

```

```

int n = 0;

Console.WriteLine($"\\nРяд для e^{x}:");
Console.WriteLine($"[{n}]: 1 = {element:F15}");

while (Math.Abs(element) >= eps)
{
    n++;
    element *= x / n;
    sum += element;
    Console.WriteLine($"[{n}]: x^{n} / {n}! = {element:F15}");
}

Console.WriteLine($"e^{x} ≈ {sum:F8}");
Console.WriteLine($"Использовано членов: {n + 1}");

static void CalculateNthTerm()
{
    double x = ReadDouble("Введите x:");
    int n = ReadInt("Введите номер члена n (n ≥ 0): ", min: 0);

    double term = 1.0;
    for (int i = 1; i <= n; i++)
        term *= x / i;

    if (n == 0)
        Console.WriteLine("0-й член: 1");
    else
        Console.WriteLine($"{n}-й член: x^{n} / {n}! = {term:F15}");
}

static double ReadDouble(string input)
{
    while (true)
    {
        Console.Write(input);
        if (double.TryParse(Console.ReadLine(), out double value))
            return value;
        Console.WriteLine("Ошибка: введите корректное число.");
    }
}

static int ReadInt(string input, int min = int.MinValue)
{
    while (true)
    {
        Console.Write(input);
        if (int.TryParse(Console.ReadLine(), out int value) && value >= min)
            return value;
        Console.WriteLine($"Ошибка: введите целое число ≥ {min}.");
    }
}
}

```

Ответ:

Реализуйте программный продукт средствами языка C# со следующим функционалом:
Вычисление значения функции $f(x)$ (соответствующей вашему варианту) с помощью ряда Маклорена с заданной точностью ϵ (ϵ и x вводятся с клавиатуры, $\epsilon < 0.01$);
Вычисление n -го члена ряда (n и x вводятся с клавиатуры). Программа вычисления e^x с помощью ряда Маклорена

Выберите действие:

1. Вычислить e^x с заданной точностью ϵ
2. Вычислить n -й член ряда
3. Выйти

Ваш выбор: 1

Введите x : 1

Введите точность ϵ ($0 < \epsilon < 0,01$): 0,001

Ряд для e^1 :

```
[0]: 1 = 1,000000000000000  
[1]: x^1 / 1! = 1,000000000000000  
[2]: x^2 / 2! = 0,500000000000000  
[3]: x^3 / 3! = 0,166666666666667  
[4]: x^4 / 4! = 0,041666666666667  
[5]: x^5 / 5! = 0,008333333333333  
[6]: x^6 / 6! = 0,001388888888889  
[7]: x^7 / 7! = 0,000198412698413
```

$e^1 \approx 2,71825397$

Использовано членов: 8

Выберите действие:

1. Вычислить e^x с заданной точностью ϵ
2. Вычислить n -й член ряда
3. Выйти

Ваш выбор: 2

Введите x : 2

Введите номер члена n ($n \geq 0$): 2

2-й член: $x^2 / 2! = 2,000000000000000$

Выберите действие:

1. Вычислить e^x с заданной точностью ϵ
2. Вычислить n -й член ряда
3. Выйти

Ваш выбор:

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

```
*****
```

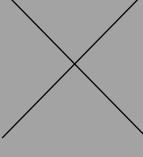
Совет: Получайте разряды числа за счет использования операций «`/`» и «`%`».

Решение:

```
using System;
namespace Task2
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("Если на билете сумма первых трёх цифр в номере билета равна\nсумме трёх последних, то этот билет считается счастливым. Напишите\nпрограмму, которая получала бы на вход шестизначный номер билета и\nвыводила, счастливый это билет или нет. К примеру: билеты 777 777\nи\n255 642 — счастливые, а 123 456 — нет.\nИспользовать при решении задачи можно только простые\nбазовые\nтипы (т.е. использование массивов, строк и коллекций запрещено,\nдолжно обрабатываться\nименно число).\n");
            long ticketNumber = ReadTicketNumber();
            int d1 = (int)(ticketNumber / 100000) % 10;
            int d2 = (int)(ticketNumber / 10000) % 10;
```

```
int d3 = (int)(ticketNumber / 1000) % 10;
int d4 = (int)(ticketNumber / 100) % 10;
int d5 = (int)(ticketNumber / 10) % 10;
int d6 = (int)(ticketNumber % 10);
int sumFirst = d1 + d2 + d3;
int sumLast = d4 + d5 + d6;
bool isLucky = sumFirst == sumLast;
Console.WriteLine(isLucky);
}
static long ReadTicketNumber()
{
    long number;
    while (true)
    {
        Console.Write("Введите номер билета: ");
        string? input = Console.ReadLine();
        if (long.TryParse(input, out number) && number >= 100000 && number <= 999999)
        {
            return number;
        }
        Console.WriteLine("Ошибка: введите шестизначное число.");
    }
}
```

Ответ:

 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).

Введите номер билета: 123006
True

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;  
  
namespace Task3  
{  
    class Program  
    {  
        static void Main()  
        {
```

```

Console.WriteLine("Пользователь вводит числа M и N. \r\nНапишите программу,
которая\r\nпреобразует дробь M/N к несократимому виду и выдаёт получившийся\r\nрезультат. ");
int numerator = ReadInteger("Введите числитель: ");
int denominator = ReadInteger("Введите знаменатель: ");
if (denominator == 0)
{
    Console.WriteLine("Ошибка: знаменатель не может быть равен 0.");
    return;
}
int gcd = GCD(Math.Abs(numerator), Math.Abs(denominator));
numerator /= gcd;
denominator /= gcd;
if (denominator < 0)
{
    numerator = -numerator;
    denominator = -denominator;
}
else if (numerator == 0) {
    Console.WriteLine($"Результат: 0");
}
else {
    Console.WriteLine($"Результат: {numerator} / {denominator}");
}
static int ReadInteger(string input)
{
    int value;
    while (true)
    {
        Console.Write(input);
        if (int.TryParse(Console.ReadLine(), out value))
        {
            return value;
        }
        Console.WriteLine("Ошибка: введите целое число.");
    }
}
static int GCD(int a, int b)
{
    while (b != 0)
    {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
}

```

Ответ:

Пользователь вводит числа M и N.
Напишите программу, которая
преобразует дробь M/N к несократимому виду и выдаёт получившийся
результат.
Введите числитель: 45
Введите знаменатель: 5455
Результат: 9 / 1091

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;
namespace Task4
{
    class Program
    {
        static void Main()
        {
            Console.WriteLine("\tНапишите программу, которая угадывает число, задуманное\r\nпользователем.\r\nЧисло загадывается в диапазоне от 0 до 63. Программа\r\nзадаёт вопросы вида «Ваше число больше такого-то?» и на основе\r\nответов пользователя («да-1» или «нет-0») угадывает число.\r\n\tАлгоритм, должен давать\r\nответ за семь вопросов. ");
            Console.WriteLine("Загадайте число от 0 до 63. Я попробую его угадать.");
            Console.WriteLine("Отвечайте:");
            Console.WriteLine(" 1 — если 'да' (ваше число больше X)");
            Console.WriteLine(" 0 — если 'нет' (ваше число меньше или равно X)");
            Console.WriteLine(" 2 — если 'да, это мое число'");
            Console.WriteLine();
            int left = 0;
            int right = 63;

            while (left <= right)
            {
                int mid = (left + right) / 2;
                Console.Write($"Ваше число больше {mid} ? ");
                int answer = ReadAnswer();

                if (answer == 2)
                {
                    Console.WriteLine($"Ваше число: {mid}");
                    return;
                }
                else if (answer == 1)
                {
                    left = mid + 1;
                }
                else if (answer == 0)
                {
                    right = mid - 1;
                }
            }
            Console.WriteLine("Вы, кажется, врёте. Нет числа, которое соответствует вашим ответам.");
        }
        static int ReadAnswer()
        {
            int answer;
            while (true)
            {
                string? input = Console.ReadLine();
                if (int.TryParse(input, out answer) && (answer == 0 || answer == 1 || answer == 2))
                {
                    return answer;
                }
            }
        }
    }
}
```

```
        }
        Console.WriteLine("Неверный ввод. Введите 0 (нет), 1 (да), или 2 (это моё число): ");
    }
}
```

Ответ:

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.
Загадайте число от 0 до 63. Я попробую его угадать.

Отвечайте:

- 1 - если 'да' (ваше число больше X)
- 0 - если 'нет' (ваше число меньше или равно X)
- 2 - если 'да, это моё число'

Ваше число больше 31? 1
Ваше число больше 47? 1
Ваше число больше 55? 1
Ваше число больше 59? 1
Ваше число больше 61? 1
Ваше число больше 62? 1
Ваше число больше 63? 2
Ваше число: 63

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 300 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:

Ведите количество воды в мл: 30

Ведите 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;
namespace Task5
{
    class Program
    {
```

```

const int WATER_FOR_AMERICANO = 300;
const int PRICE_AMERICANO = 150;
const int WATER_FOR_LATTE = 30;
const int MILK_FOR_LATTE = 270;
const int PRICE_LATTE = 170;

static void Main()
{
    Console.WriteLine("\tКофейный аппарат может готовить два напитка: американо и латте.\r\nДля
американо требуется 300 мл воды (цена 150 рублей), а для латте 30\r\nмл воды и 270 мл молока (цена 170
рублей).\r\n\tНапишите программу, которая спрашивает у пользователя (это\r\ndействие программа делает
один раз в начале работы), сколько всего\r\n9\r\nмиллилитров молока и воды залито в кофейный
аппарат.\r\n\tПосле чего начинает обслуживание пользователей, запрашивается,\r\nкакой напиток хочет
заказать посетитель. Пользователь выбирает один\r\nиз двух напитков, программа отвечает одним из трёх
вариантов: «Ваш\r\nнапиток готов», «Не хватает воды» или «Не хватает молока», после чего\r\nпереходит к
обслуживанию следующего посетителя. Если молока и воды\r\nне хватает ни на один вид напитка,
программа выдаёт отчёт и\r\nзавершается.\r\n\tВ отчёте должно быть написано, что ингредиенты подошли
к\r\nконцу, должен быть указан остаток воды и молока в машине, должно\r\nбыть указано, сколько всего
было приготовлено чашек американо и латте\r\nпза эту смену и итоговый заработок аппарата. ");
    Console.Write("\n Введите количество воды в мл: ");
    int totalWater = ReadInt();
    Console.Write(" Введите количество молока в мл: ");
    int totalMilk = ReadInt();

    int cupsAmericano = 0;
    int cupsLatte = 0;
    int totalIncome = 0;

    while (true)
    {
        bool canMakeAmericano = totalWater >= WATER_FOR_AMERICANO;
        bool canMakeLatte = totalWater >= WATER_FOR_LATTE && totalMilk >= MILK_FOR_LATTE;

        if (!canMakeAmericano && !canMakeLatte)
        {
            Console.WriteLine(" Недостаточно ингредиентов для приготовления любого напитка.");
            break;
        }
        Console.WriteLine(" Доступные напитки:");
        if (canMakeAmericano)
        {
            Console.WriteLine(" 1 — Американо");
        }
        if (canMakeLatte)
        {
            Console.WriteLine(" 2 — Латте");
        }

        Console.Write(" Ваш выбор: ");
        int choice = ReadInt();
        if (choice == 1 && canMakeAmericano)
        {
            totalWater -= WATER_FOR_AMERICANO;
            cupsAmericano++;
            totalIncome += PRICE_AMERICANO;
            Console.WriteLine(" Ваш напиток готов.");
        }
        else if (choice == 2 && canMakeLatte)
        {
            totalWater -= WATER_FOR_LATTE;
            totalMilk -= MILK_FOR_LATTE;
            cupsLatte++;
            totalIncome += PRICE_LATTE;
            Console.WriteLine(" Ваш напиток готов.");
        }
        else
        {
    
```

```
        Console.WriteLine(" Недоступный или неверный выбор. Пожалуйста, выберите из доступных напитков.");
    }

    Console.WriteLine();
}
Console.WriteLine("*Отчёт*");
Console.WriteLine(" Ингредиентов осталось:");
Console.WriteLine($" Вода: {totalWater} мл");
Console.WriteLine($" Молоко: {totalMilk} мл");
Console.WriteLine($" Кружек американо приготовлено: {cupsAmericano}");
Console.WriteLine($" Кружек латте приготовлено: {cupsLatte}");
Console.WriteLine($" Итого: {totalIncome} рублей.");
}

static int ReadInt()
{
    int value;
    while (true)
    {
        string input = Console.ReadLine();
        if (int.TryParse(input, out value))
        {
            return value;
        }
        Console.Write(" Некорректный ввод. Введите число: ");
    }
}
```

Ответ:

Кофейный аппарат может готовить два напитка: американо и латте. Для американо требуется 300 мл воды (цена 150 рублей), а для латте 300 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 9

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 заработка аппарата.

Введите количество воды в мл: 1000

Введите количество молока в мл: 1000

Доступные напитки:

1 - Американо

2 - Латте

Ваш выбор: 2

Ваш напиток готов.

Доступные напитки:

1 - Американо

2 - Латте

Ваш выбор: 2

Ваш напиток готов.

Доступные напитки:

1 - Американо

2 - Латте

Ваш выбор: 2

Ваш напиток готов.

Доступные напитки:

1 - Американо

Ваш выбор: 1

Ваш напиток готов.

Доступные напитки:

1 - Американо

Ваш выбор: 1

Ваш напиток готов.

Доступные напитки:

1 - Американо

Ваш выбор: 1

Ваш напиток готов.

Недостаточно ингредиентов для приготовления любого напитка.

Отчёт

Ингредиентов осталось:

Вода: 10 мл

Молоко: 190 мл

Кружек американо приготовлено: 3

Кружек латте приготовлено: 3

Итого: 960 рублей.

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 становится 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

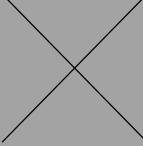
Решение:

```
using System;

namespace Task6
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("\tВ чашку Петри кладут N бактерий и добавляют X капель антибиотика (N и X вводятся с клавиатуры).\n\nИзвестно, 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.\n\nПользователь 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.\n\nЦикл не должен быть бесконечным (после того как количество антибиотики или бактерий становится равным нулю выполнение программы должно быть завершено).");
            Console.Write("\nВведите количество бактерий: ");
            int bacteria = ReadInt();
            Console.Write("Введите количество антибиотика: ");
            int antibioticDrops = ReadInt();
            int hour = 1;
            int killsPerHour = 10;
            while (bacteria > 0 && killsPerHour > 0 && antibioticDrops > 0)
            {
                bacteria *= 2;
                int killed = Math.Min(bacteria, killsPerHour * antibioticDrops);
                bacteria -= killed;
                Console.WriteLine($"После {hour} часа бактерий осталось {bacteria}");
                killsPerHour--;
                hour++;
                if (bacteria <= 0)
                {
                    break;
                }
            }

            static int ReadInt()
            {
                int value;
                while (true)
                {
                    if (int.TryParse(Console.ReadLine(), out value))
                    {
                        return value;
                    }
                    Console.Write("Некорректный ввод. Введите число: ");
                }
            }
        }
    }
}
```

Ответ:

 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 становится равным нулю выполнение программы должно быть завершено).

Введите количество бактерий: 110

Введите количество антибиотика: 13

После 1 часа бактерий осталось 90

После 2 часа бактерий осталось 63

После 3 часа бактерий осталось 22

После 4 часа бактерий осталось 0

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 $a \times b$ 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 $(a + 2d) \times (b + 2d)$ метров.

Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером $w \times h$ 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

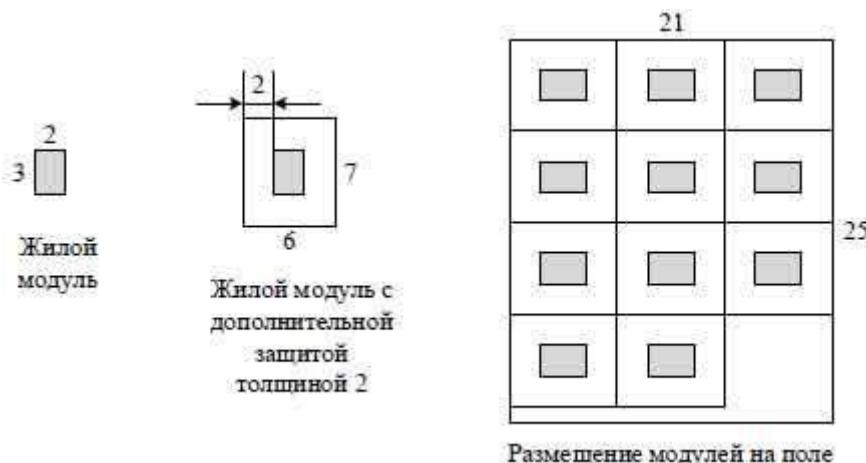
Введите b: 2

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.



Решение:

```
using System;

namespace Task7
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Требуется написать программу, которая по заданным количеству n\\ при 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\\ пклавиатуры)");
            Console.Write("Введите n: ");
            int n = ReadInt();
            Console.Write("Введите a: ");
            int a = ReadInt();
            Console.Write("Введите b: ");
            int b = ReadInt();
            Console.Write("Введите w: ");
            int w = ReadInt();
            Console.Write("Введите h: ");
            int h = ReadInt();

            int maxD = Math.Min(w, h) / 2;
            int result = -1;
            for (int d = 0; d <= maxD; d++)
            {
                int moduleW1 = a + 2 * d;
                int moduleH1 = b + 2 * d;
                int moduleW2 = b + 2 * d;
                int moduleH2 = a + 2 * d;
                if (CanPlaceModulesMixed(n, moduleW1, moduleH1, moduleW2, moduleH2, w, h))
                    result = d;
            }
            Console.WriteLine(result);
        }

        static int ReadInt()
        {
            string s;
            int result;
            while (!int.TryParse(s = Console.ReadLine(), out result));
            return result;
        }

        static bool CanPlaceModulesMixed(int n, int moduleW1, int moduleH1, int moduleW2, int moduleH2, int w, int h)
        {
            if ((moduleW1 + moduleW2) * n > w || (moduleH1 + moduleH2) * n > h)
                return false;
            if ((moduleW1 + moduleW2) * n < w || (moduleH1 + moduleH2) * n < h)
                return true;
            if ((moduleW1 + moduleW2) * n == w && (moduleH1 + moduleH2) * n == h)
                return true;
            if ((moduleW1 + moduleW2) * n == w && (moduleH1 + moduleH2) * n < h)
                return true;
            if ((moduleW1 + moduleW2) * n < w && (moduleH1 + moduleH2) * n == h)
                return true;
            return false;
        }
    }
}
```

```

        {
            result = d;
        }
    else
    {
        break;
    }
}

if (result == -1)
{
    Console.WriteLine("Ответ: невозможно разместить модули даже без дополнительной защиты.");
}
else
{
    Console.WriteLine($"Ответ: максимальная толщина защиты d = {result}");
}
}

static bool CanPlaceModulesMixed(int n, int w1, int h1, int w2, int h2, int W, int H)
{
    if (GridFit(n, w1, h1, W, H) || GridFit(n, w2, h2, W, H))
    {
        return true;
    }

    for (int k = 1; k < n; k++)
    {
        int countType2 = n - k;
        for (int splitW = 1; splitW < W; splitW++)
        {
            if (GridFit(k, w1, h1, splitW, H) && GridFit(countType2, w2, h2, W - splitW, H))
            {
                return true;
            }
        }
        for (int splitW = 1; splitW < W; splitW++)
        {
            if (GridFit(k, w2, h2, splitW, H) && GridFit(countType2, w1, h1, W - splitW, H))
            {
                return true;
            }
        }
        for (int splitH = 1; splitH < H; splitH++)
        {
            if (GridFit(k, w1, h1, W, splitH) && GridFit(countType2, w2, h2, W, H - splitH))
            {
                return true;
            }
        }
        for (int splitH = 1; splitH < H; splitH++)
        {
            if (GridFit(k, w2, h2, W, splitH) && GridFit(countType2, w1, h1, W, H - splitH))
            {
                return true;
            }
        }
    }
    return false;
}

static bool GridFit(int n, int wMod, int hMod, int W, int H)
{
    if (wMod > W || hMod > H) return false;
    int count = (W / wMod) * (H / hMod);
    return count >= n;
}

static int ReadInt()

```

```
{  
    while (true)  
    {  
        if (int.TryParse(Console.ReadLine(), out int value))  
            return value;  
        Console.Write("Некорректный ввод. Пожалуйста, введите целое число: ");  
    }  
}
```

Ответ:

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)

Введите n : 12
Введите a : 2
Введите b : 3
Введите w : 15
Введите h : 11
Ответ: максимальная толщина защиты $d = 0$