

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ИБ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Объектно-ориентированное программирование»
Тема: Использование базовых языковых конструкций

Студент гр. 3363

Минко Д.А.

Преподаватель

Новакова Н.Е.

Санкт-Петербург
2024

Цель работы.

Цель данной лабораторной работы освоение применения базовых языковых конструкций, работы с исключениями и добавление функциональности в программы с использованием среды Microsoft Visual Studio. В ходе выполнения работы необходимо реализовать преобразование чисел в формат "месяц – день", обработку ошибок ввода через исключения, а также модификацию программы для поддержки високосных годов.

Ход работы.

1. Конвертирование числа, соответствующего дню года, в пару «месяц – день».

Была открыта среда Visual Studio 2022, в меню File выбран пункт New, затем Project. Из списка шаблонов выбрано Console Application с названием проекта WhatDay и указан путь C:\Labs\Lab3. Использован код из файла WhatDay1.cs, содержащий объявление класса WhatDay и коллекцию дней по месяцам. В метод Main добавлена конструкция Console.WriteLine для ввода числа, соответствующего дню года, объявлены переменные line и dayNum, инициализированные методами Console.ReadLine и int.Parse. Программа скомпилирована, исправлены ошибки, добавлены переменные и удалены комментарии из конструкций if и switch для всех месяцев. Дописаны команды вывода dayNum и monthName, программа снова скомпилирована и сохранена (рис. 1).

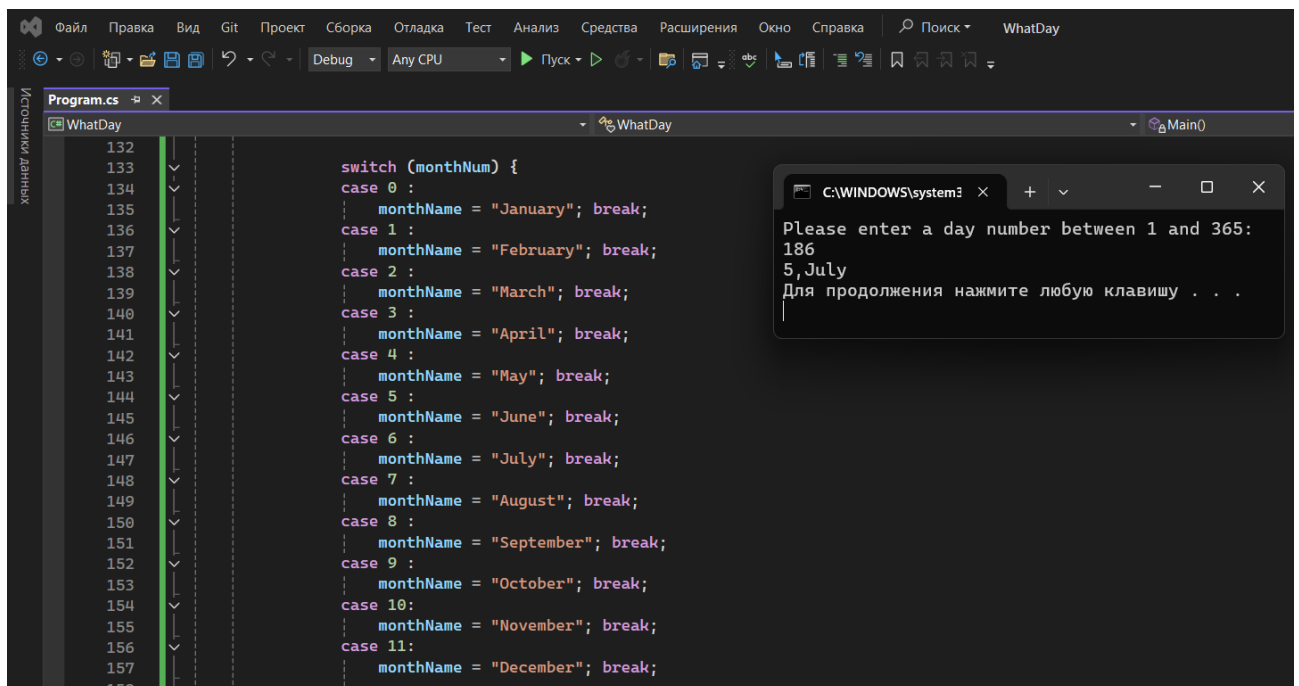


Рисунок 1 – Программа с неоптимизированным кодом

Была модифицирована программа, в которой конструкция switch была заменена на перечисление MonthName, содержащее названия 12

месяцев. Конструкция switch была закомментирована, вместо неё была объявлена переменная temp как перечисление и инициализирована на основе переменной monthNum. Для получения названия месяца использовалось выражение temp.ToString(). Программа скомпилирована, ошибки исправлены, результат сохранён.

Затем 12 операторов if были заменены на цикл foreach, который итерационно обрабатывает коллекцию DaysInMonths. В теле цикла была написана конструкция if, использующая переменную daysInMonth вместо числовых значений. Метка End была закомментирована, оператор goto заменён на break. Программа вновь скомпилирована, ошибки исправлены, и результат сохранён (рис. 2).

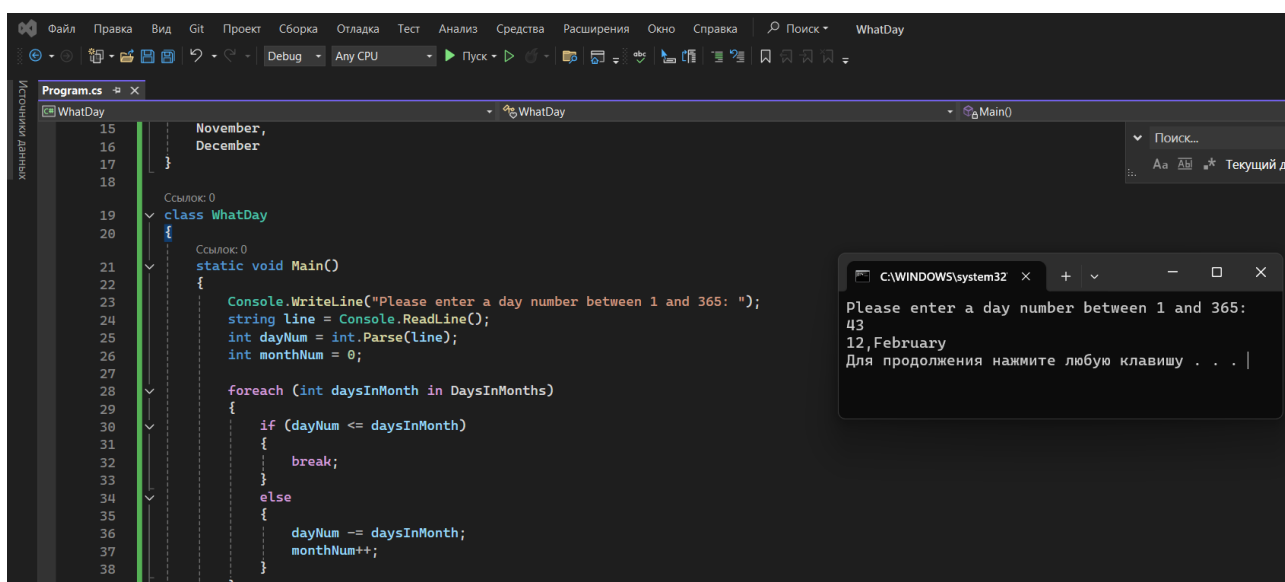


Рисунок 2 – Программа с неоптимизированным кодом

2. Использование исключений (рис. 3).

В программу, написанную в предыдущем упражнении, были внесены изменения для добавления проверки введённого пользователем числа, соответствующего дню года. Если введённое значение оказалось меньше 1 или больше 365, было реализовано исключение с помощью конструкции throw. Для обработки этого исключения был добавлен блок try-catch, в котором программа перехватывает исключение типа ArgumentOutOfRangeException. В блоке catch

выводится сообщение об ошибке, информирующее пользователя о некорректном вводе. Программа была скомпилирована, ошибки исправлены, и результат сохранён.

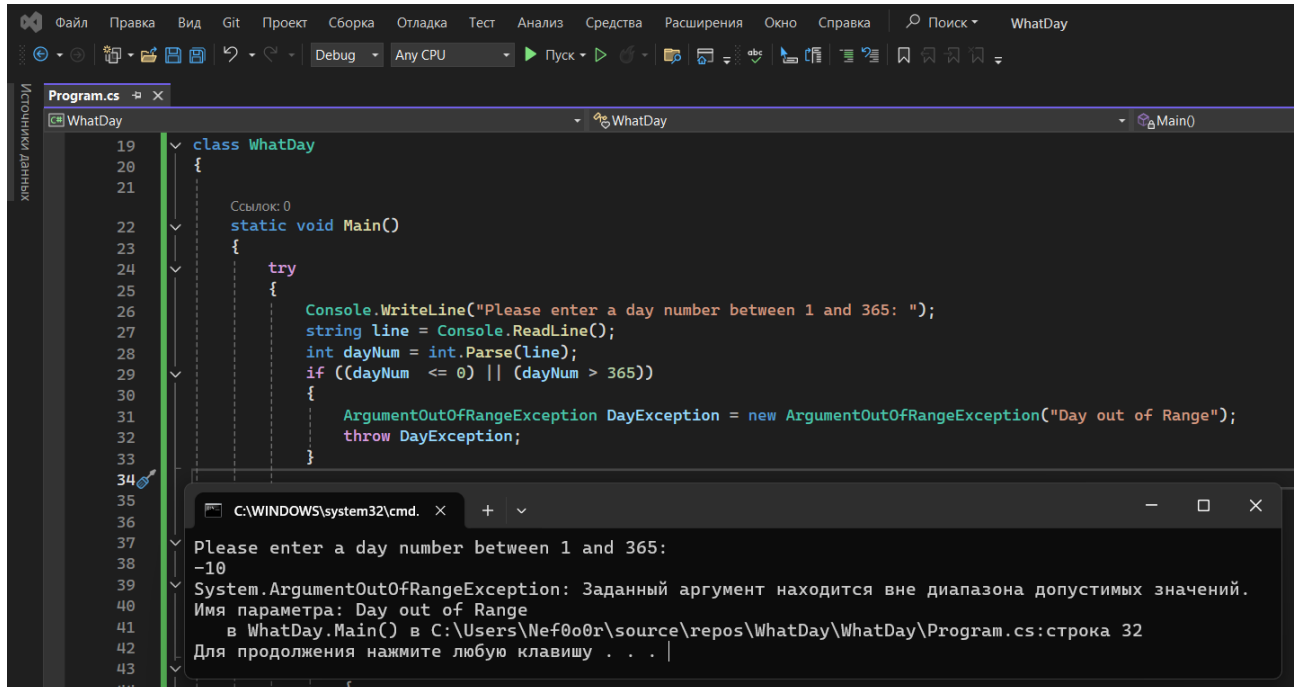


Рисунок 3 – Добавление и обработка исключений

3. Ввод в программу дополнительной функциональности

В проект, созданный в предыдущем упражнении, были внесены следующие изменения. В начале метода Main, перед запросом номера дня, добавлено приглашение пользователю ввести год. Была объявлена переменная line типа string и инициализирована с помощью метода Console.ReadLine(). Затем добавлено выражение для объявления переменной yearNum типа int, которая была инициализирована с помощью метода int.Parse(line).

После этого была объявлена переменная isLeapYear типа bool, которая была инициализирована с помощью проверки, определяющей, является ли введённый год високосным. Программа была скомпилирована, ошибки откорректированы, и результат сохранён (рис. 4).

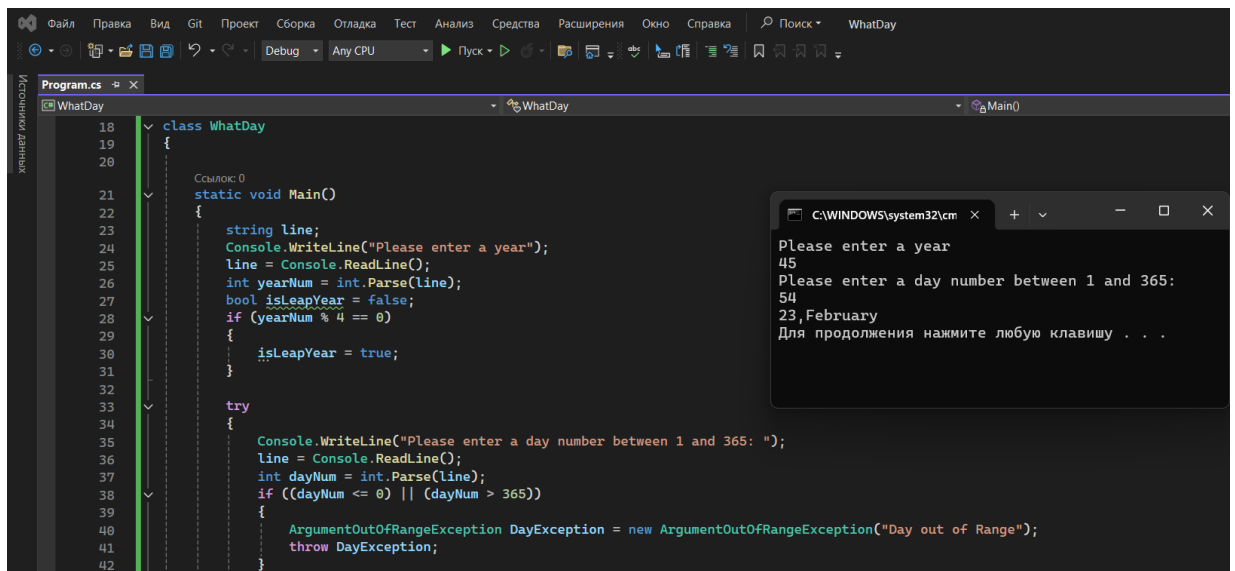


Рисунок 4 – Добавление проверку на високосность года

После объявления переменной `isLeapYear` была добавлена переменная `maxDayNum` типа `int`, значение которой инициализировалось как 365 или 366 в зависимости от того, является ли год високосным (значение переменной `isLeapYear` — `true` или `false`). Затем были внесены изменения в команду `Console.WriteLine`, которая приглашала пользователя ввести число от 1 до 365 — вместо константы теперь использовалась переменная `maxDayNum`, чтобы учесть високосные годы. Программа была скомпилирована, исправлены ошибки, и результат сохранён (рис. 5).

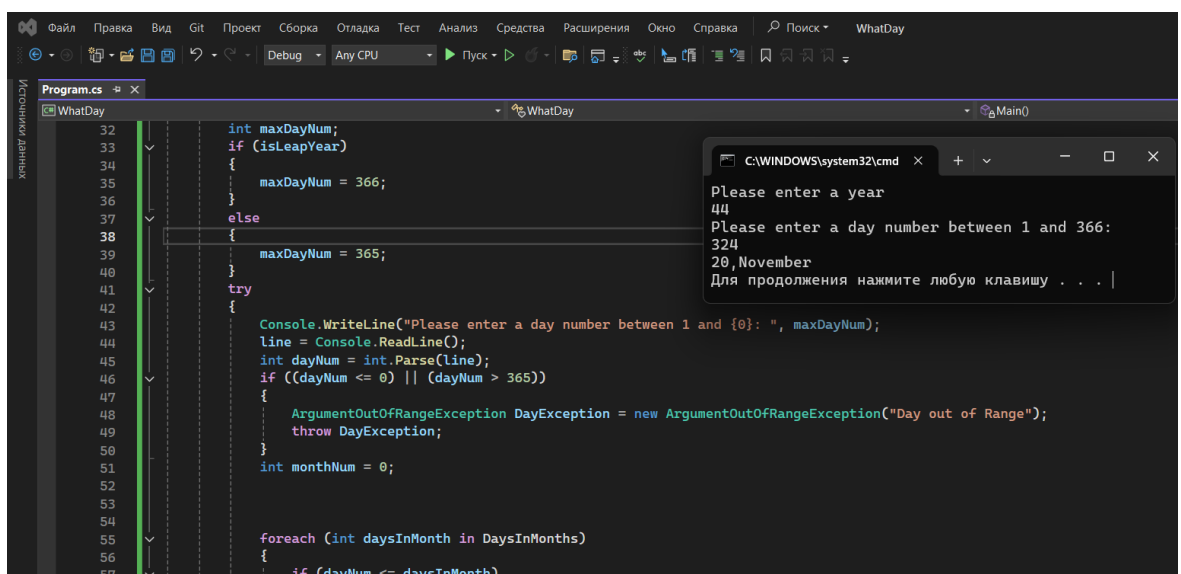


Рисунок 5 – Замена константы при выводе текста на переменную

Процедура вычисления пары «месяц – день» была откорректирована для учета високосного года. После конструкции if, проверяющей день, и объявления переменной monthNum, была добавлена конструкция if-else, где булевское выражение — это переменная isLeapYear. Конструкция foreach была преобразована так, чтобы она присутствовала в обеих частях выражения if-else. Для обработки високосного года использована коллекция DaysInLeapMonth, которая аналогична DaysInMonths, но для февраля содержит 29 дней. Программа была скомпилирована, ошибки исправлены, и результат сохранён (рис. 6).

```

Program.cs
WhatDay
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103

    daynum -= daysInMonth;
    monthNum++;
}
}

MonthName temp = (MonthName)monthNum;
string monthName = temp.ToString();
Console.WriteLine("{0},{1}", dayNum, monthName);
}
catch (System.Exception caught)
{
    Console.WriteLine(caught.ToString());
}

static System.Collections.ICollection DaysInMonths
= new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
static System.Collections.ICollection DaysInLeapMonths
= new int[12] { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

```

C:\WINDOWS\system32 x + - □ x

Please enter a year
32
Please enter a day number between 1 and 366:
121
30, April
Для продолжения нажмите любую клавишу . . . |

Рисунок 6 – Учет високосности года при окончательных подсчётах

Вывод.

В ходе выполнения лабораторной работы были освоены базовые языковые конструкции, такие как условные операторы, циклы и перечисления, а также методы обработки исключений. Были реализованы программы для преобразования числового значения в пару "месяц – день" и добавлена поддержка високосного года. Эти задачи позволили закрепить навыки работы с переменными, операторами управления потоком и коллекциями. Работа также способствовала развитию навыков выявления и исправления ошибок в процессе компиляции и выполнения программ.

ИСХОДНЫЙ КОД

Упражнение 1 (пункт 15):

```
using System;
enum MonthName
{
    January,
    February,
    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{
    static void Main()
    {
        Console.WriteLine("Please enter a day number between 1 and 365: ");
        string line = Console.ReadLine();
        int dayNum = int.Parse(line);
        int monthNum = 0;

        foreach (int daysInMonth in DaysInMonths)
        {
            if (dayNum <= daysInMonth)
            {
                break;
            }
            else
            {
                dayNum -= daysInMonth;
                monthNum++;
            }
        }
        /*
        if (dayNum <= 31) { // January
            goto End;
        } else {
            dayNum -= 31;
            monthNum++;
        }

        if (dayNum <= 28) { // February
            goto End;
```

```

    } else {
        dayNum -= 28;
        monthNum++;
    }

    if (dayNum <= 31) { // March
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }

    if (dayNum <= 30) { // April
        goto End;
    } else {
        dayNum -= 30;
        monthNum++;
    }

    if (dayNum <= 31) { // May
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }

    if (dayNum <= 30) { // June
        goto End;
    } else {
        dayNum -= 30;
        monthNum++;
    }

    if (dayNum <= 31) { // July
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }

    if (dayNum <= 31) { // August
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }

    if (dayNum <= 30) { // September
        goto End;
    } else {
        dayNum -= 30;

```

```

        monthNum++;
    }

    if (dayNum <= 31) { // October
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }
    if (dayNum <= 31) { // November
        goto End;
    } else {
        dayNum -= 30;
        monthNum++;
    }
    if (dayNum <= 31) { // December
        goto End;
    } else {
        dayNum -= 31;
        monthNum++;
    }
    */
End:
    //string MonthName;
    // MonthName temp = (MonthName)monthNum;
    //string monthName = temp.ToString();
    MonthName temp = (MonthName)monthNum;
    string monthName = temp.ToString();

    /*
        switch (monthNum) {
        case 0 :
            monthName = "January"; break;
        case 1 :
            monthName = "February"; break;
        case 2 :
            monthName = "March"; break;
        case 3 :
            monthName = "April"; break;
        case 4 :
            monthName = "May"; break;
        case 5 :
            monthName = "June"; break;
        case 6 :
            monthName = "July"; break;
        case 7 :
            monthName = "August"; break;
        case 8 :
            monthName = "September"; break;
        case 9 :
            monthName = "October"; break;
        case 10:

```

```

        monthName = "nOVEMBER"; break;
    case 11:
        monthName = "December"; break;
    default:
        monthName = "not done yet"; break;
    }
    */
    Console.WriteLine("{0},{1}", dayNum, monthName);
}
static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```

Упражнение 1 (пункт 21):

```

using System;
enum MonthName
{
    January,
    February,
    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{
    static void Main()
    {
        Console.WriteLine("Please enter a day number between 1 and 365: ");
        string line = Console.ReadLine();
        int dayNum = int.Parse(line);
        int monthNum = 0;

        foreach (int daysInMonth in DaysInMonths)
        {
            if (dayNum <= daysInMonth)
            {
                break;
            }
            else
            {
                dayNum -= daysInMonth;
                monthNum++;
            }
        }
    }
}

```

```

    }
    MonthName temp = (MonthName)monthNum;
    string monthName = temp.ToString();
    Console.WriteLine("{0},{1}", dayNum, monthName);
}
static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```

Упражнение 2:

```

using System;
enum MonthName
{
    January,
    February,
    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Please enter a day number between 1 and 365: ");
            string line = Console.ReadLine();
            int dayNum = int.Parse(line);
            if ((dayNum <= 0) || (dayNum > 365))
            {
                ArgumentOutOfRangeException DayException = new
ArgumentOutOfRangeException("Day out of Range");
                throw DayException;
            }

            int monthNum = 0;

            foreach (int daysInMonth in DaysInMonths)
            {
                if (dayNum <= daysInMonth)
                {
                    break;

```

```

        }
        else
        {
            dayNum -= daysInMonth;
            monthNum++;
        }
    }
    MonthName temp = (MonthName)monthNum;
    string monthName = temp.ToString();
    Console.WriteLine("{0},{1}", dayNum, monthName);
}
catch (System.Exception caught)
{
    Console.WriteLine(caught.ToString());
}

}
static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```

Упражнение 3 (пункт 7):

```

using System;
enum MonthName
{
    January,
    February,
    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{
    static void Main()
    {
        string line;
        Console.WriteLine("Please enter a year");
        line = Console.ReadLine();
        int yearNum = int.Parse(line);
        bool isLeapYear = false;
        if (yearNum % 4 == 0)

```

```

    {
        isLeapYear = true;
    }

    try
    {
        Console.WriteLine("Please enter a day number between 1 and 365: ");
        line = Console.ReadLine();
        int dayNum = int.Parse(line);
        if ((dayNum <= 0) || (dayNum > 365))
        {
            ArgumentOutOfRangeException DayException = new
ArgumentOutOfRangeException("Day out of Range");
            throw DayException;
        }
        int monthNum = 0;

        foreach (int daysInMonth in DaysInMonths)
        {
            if (dayNum <= daysInMonth)
            {
                break;
            }
            else
            {
                dayNum -= daysInMonth;
                monthNum++;
            }
        }
        MonthName temp = (MonthName)monthNum;
        string monthName = temp.ToString();
        Console.WriteLine("{0},{1}", dayNum, monthName);
    }
    catch (System.Exception caught)
    {
        Console.WriteLine(caught.ToString());
    }

}

static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```

Упражнение 3 (пункт 10):

```

using System;
enum MonthName
{
    January,
    February,

```

```

    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{

    static void Main()
    {
        string line;
        Console.WriteLine("Please enter a year");
        line = Console.ReadLine();
        int yearNum = int.Parse(line);
        bool isLeapYear = false;
        if (yearNum % 4 == 0)
        {
            isLeapYear = true;
        }
        int maxDayNum;
        if (isLeapYear)
        {
            maxDayNum = 366;
        }
        else
        {
            maxDayNum = 365;
        }
        try
        {
            Console.WriteLine("Please enter a day number between 1 and {0}: ", maxDayNum);
            line = Console.ReadLine();
            int dayNum = int.Parse(line);
            if ((dayNum <= 0) || (dayNum > 365))
            {
                ArgumentOutOfRangeException DayException = new
ArgumentOutOfRangeException("Day out of Range");
                throw DayException;
            }
            int monthNum = 0;

            foreach (int daysInMonth in DaysInMonths)
            {

```



```

        if (dayNum <= daysInMonth)
        {
            break;
        }
        else
        {
            dayNum -= daysInMonth;
            monthNum++;
        }
    }
    MonthName temp = (MonthName)monthNum;
    string monthName = temp.ToString();
    Console.WriteLine("{0},{1}", dayNum, monthName);
}
catch (System.Exception caught)
{
    Console.WriteLine(caught.ToString());
}

}
static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```

Упражнение 3 (пункт 15):

```

using System;
using System.Diagnostics.Eventing.Reader;
enum MonthName
{
    January,
    February,
    March,
    April,
    May,
    June,
    July,
    August,
    September,
    October,
    November,
    December
}

class WhatDay
{
    static void Main()
    {
        string line;
        Console.WriteLine("Please enter a year");
        line = Console.ReadLine();
    }
}

```

```

int yearNum = int.Parse(line);
bool isLeapYear = false;
if (yearNum % 4 == 0)
{
    isLeapYear = true;
}
int maxDayNum;
if (isLeapYear)
{
    maxDayNum = 366;
}
else
{
    maxDayNum = 365;
}
try
{
    Console.WriteLine("Please enter a day number between 1 and {0}: ", maxDayNum);
    line = Console.ReadLine();
    int dayNum = int.Parse(line);
    if ((dayNum <= 0) || (dayNum > 365))
    {
        ArgumentOutOfRangeException DayException = new
ArgumentOutOfRangeException("Day out of Range");
        throw DayException;
    }
    int monthNum = 0;

    if (isLeapYear)
    {
        foreach (int daysInMonth in DaysInLeapMonths)
        {
            if (dayNum <= daysInMonth)
            {
                break;
            }
            else
            {
                dayNum -= daysInMonth;
                monthNum++;
            }
        }
    }
    else
    {
        foreach (int daysInMonth in DaysInMonths)
        {
            if (dayNum <= daysInMonth)
            {
                break;
            }
            else

```

```

        {
            dayNum -= daysInMonth;
            monthNum++;
        }
    }
}

```

```

        MonthName temp = (MonthName)monthNum;
        string monthName = temp.ToString();
        Console.WriteLine("{0},{1}", dayNum, monthName);
    }
    catch (System.Exception caught)
    {
        Console.WriteLine(caught.ToString());
    }
}

static System.Collections.ICollection DaysInMonths
    = new int[12] { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
static System.Collections.ICollection DaysInLeapMonths
    = new int[12] { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
}

```