



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ ИУК «Информатика и управление»**

**КАФЕДРА ИУК2 «Информационные системы и сети»**

**Практическая работа 1.4**  
**«Обработка исключительных ситуаций в С#. Стандартные**  
**классы: символы, строки и организация потокового ввода**  
**вывода»**

**ДИСЦИПЛИНА: «Объектно-ориентированное программирование»**

Выполнил: студент гр. ИУК4-21Б

  
(подпись)

( Суриков Н.С )  
(Ф.И.О.)

Проверил:

(подпись)

( Дерюгина Е. О. )  
(Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

**Цель:** сформировать навыки решения задач с применением ООП.

**Задачи:** изучить особенности работы с ООП.

## Вариант 20

### Задание:

1. Постройте таблицу значений функции  $y = f(x)$  для  $x \in [a, b]$  с шагом  $h$ . Если в некоторой точке  $x$  функция не определена, то выведите на экран сообщение об этом.

При решении данной задачи использовать вспомогательный метод  $f(x)$ , реализующий заданную функцию, а также проводить обработку возможных исключений.

$$y = \frac{1}{x + 7} + \ln(1 - |x|)$$

2А. Разработать программу, которая для заданной строки  $s$  определяет, сколько различных символов встречается в строке.

2Б. Дана строка, в которой содержится осмысленное текстовое сообщение. Слова сообщения разделяются пробелами и знаками препинания. Вывести слова сообщения в порядке возрастания их длин.

3. Дана строка, в которой содержится осмысленное текстовое сообщение. Слова сообщения разделяются пробелами и знаками препинания.

В сообщении может содержаться время в формате  $чч:мм$ . В заданном формате  $чч$  - целое число из диапазона от 00 до 24,  $мм$  - целое число из диапазона от 00 до 60 (если какая-то часть формата нарушена, то данная подстрока в качестве даты не рассматривается).

Уменьшите время на  $n$  часов.

4. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются на ту же букву, что и последнее слово.

5. Дан текстовый файл. Переписать в новый файл все его строки, удалив из них символы, стоящие на четных местах.

### Листинг:

```
1 using System;
2 using System.Text.RegularExpressions;
3 using System.IO;
4 using System.Text;
5
6
```

```

7 namespace ПРЗ
8 {
9     /*
10     Постройте таблицу значений функции  $f(x)$  для  $x[a, b]$  с шагом  $h$ 
11     Если в некоторой точке функция не определена, то выведите на экран
    сообщение об этом.
12     При решении данной задачи использовать вспомогательный метод  $f(x)$ ,
13     реализующий заданную функцию, а также проводить обработку возможных
    исключений.
14
15      $f(x) = 1/\ln(x+7) + \ln(1 - \text{abs}(x))$ 
16     */
17     class Function
18     {
19         private decimal a;
20         private decimal b;
21         private decimal h;
22         public Function(decimal a, decimal b, decimal h)
23         {
24             this.a = a;
25             this.b = b;
26             this.h = h;
27         }
28         public decimal Calculate(decimal x)
29         {
30             if (x <= -1 || x >= 1)
31             {
32                 throw new Exception($"Функция не определена в точке x =
    {x}");
33             }
34             return 1 / (decimal)Math.Log((double)(x + 7)) +
    (decimal)Math.Log(1 - Math.Abs((double)x));
35         }
36         public void TableOfValues()
37         {
38             for (decimal x = this.a; x <= this.b; x += this.h)
39             {
40                 try
41                 {
42                     Console.WriteLine($"f({x}) =
    {Calculate(x):0.00000f}");
43                 }
44                 catch (Exception e)
45                 {
46                     Console.WriteLine(e.Message);
47                 }
48             }
49         }
50     }
51
52     /*
53     Разработать программу, которая для заданной строки s:
54     определяет, сколько различных символов встречается в строке.
55
56     Дана строка, в которой содержится осмысленное текстовое сообщение.

```

```

57      Слова сообщения разделяются пробелами и знаками препинания.
58      Вывести слова сообщения в порядке возрастания их длин.
59      */
60
61      class String
62      {
63          private string s;
64          public String(string s)
65          {
66              this.s = s;
67          }
68          // A
69          public int CountOfDifferentSymbols()
70          {
71              return s.Distinct().Count();
72          }
73          // B
74          public void WordsInAscendingOrder()
75          {
76              string[] words = s.Split(new char[] { ' ', ',', '.', '!',
'? ', ': ' }, StringSplitOptions.RemoveEmptyEntries);
77              Array.Sort(words, (a, b) => a.Length.CompareTo(b.Length));
78              foreach (string word in words)
79              {
80                  Console.WriteLine(word);
81              }
82          }
83      }
84
85      /*
86      Регулярные выражения C#
87
88      Дана строка, в которой содержится осмысленное текстовое сообщение.
89      Слова сообщения разделяются пробелами и знаками препинания.
90
91      В сообщении может содержаться время в формате чч:мм.
92      В заданном формате чч - целое число из диапазона от 00 до 24,
93      мм - целое число из диапазона от 00 до 60 (если какая-то часть
формата нарушена,
94      то данная подстрока в качестве даты не рассматривается).
95      Уменьшите время на n часов.
96      */
97
98      public class TimeReducer
99      {
100          private string text;
101
102          public TimeReducer(string text)
103          {
104              this.text = text;
105          }
106
107          public string ReduceTime(int hoursToSubtract)
108          {

```

```

109         return Regex.Replace(text, @"\b(?:[01]?[d|2[0-3]]):[0-5]?[d\
b", match =>
110         {
111             string time = match.Value;
112             return SubtractHours(time, hoursToSubtract);
113         });
114     }
115
116     private string SubtractHours(string time, int hoursToSubtract)
117     {
118         string[] parts = time.Split(':');
119         int hours = int.Parse(parts[0]);
120         int minutes = int.Parse(parts[1]);
121
122         hours -= hoursToSubtract;
123
124         if (hours < 0)
125         {
126             hours += 24;
127         }
128         else if (hours >= 24)
129         {
130             hours -= 24;
131         }
132         return $"{hours:D2}:{minutes:D2}";
133     }
134 }
135
136 /*
137  Работа с двоичными файлами:
138  Создать файл, состоящий из слов.
139  Вывести на экран все слова, которые начинаются на ту же букву, что и
последнее слово.
140  */
141
142 public class WordBinaryFile
143 {
144     private string filePath;
145
146     public WordBinaryFile(string filePath)
147     {
148         this.filePath = filePath;
149     }
150
151     public void CreateFile(string[] words)
152     {
153         using (BinaryWriter writer = new
BinaryWriter(File.Open(filePath, FileMode.Create)))
154         {
155             foreach (string word in words)
156             {
157                 byte[] wordBytes = Encoding.UTF8.GetBytes(word);
158                 writer.Write(wordBytes.Length);
159                 writer.Write(wordBytes);
160             }

```

```

161     }
162 }
163
164     public void PrintWordsStartingWithLastLetter()
165     {
166         using (BinaryReader reader = new
BinaryReader(File.Open(filePath, FileMode.Open)))
167         {
168             string lastWord = "";
169             while (reader.PeekChar() != -1)
170             {
171                 int wordLength = reader.ReadInt32();
172                 byte[] wordBytes = reader.ReadBytes(wordLength);
173                 string word = Encoding.UTF8.GetString(wordBytes);
174
175                 if (lastWord.Length > 0 &&
word.StartsWith(lastWord[lastWord.Length - 1].ToString(),
StringComparison.OrdinalIgnoreCase))
176                 {
177                     Console.WriteLine(word);
178                 }
179
180                 lastWord = word;
181             }
182         }
183     }
184 }
185
186     public class TextFileProcessor
187     {
188         private string inputFilePath;
189         private string outputFilePath;
190
191         public TextFileProcessor(string inputFilePath, string
outputFilePath)
192         {
193             this.inputFilePath = inputFilePath;
194             this.outputFilePath = outputFilePath;
195         }
196
197         public void ProcessFile()
198         {
199             string[] lines = File.ReadAllLines(inputFilePath);
200
201             using (StreamWriter writer = new
StreamWriter(outputFilePath))
202             {
203                 foreach (string line in lines)
204                 {
205                     string modifiedLine = RemoveEvenPositionChars(line);
206                     writer.WriteLine(modifiedLine);
207                 }
208             }
209         }
210

```

```

211     private string RemoveEvenPositionChars(string str)
212     {
213         StringBuilder sb = new StringBuilder();
214
215         for (int i = 0; i < str.Length; i++)
216         {
217             if (i % 2 != 0)
218             {
219                 sb.Append(str[i]);
220             }
221         }
222
223         return sb.ToString();
224     }
225 }
226 class Program
227 {
228     static void Main(string[] args)
229     {
230         Function f = new Function(-10, 10, 0.2m);
231         f.TableOfValues();
232
233         String s = new String("Hello, World!");
234         Console.WriteLine($"Количество различных символов в строке:
235 {s.CountOfDifferentSymbols()}");
236         s.WordsInAscendingOrder();
237
238         string message = "Встреча в 24:50. Не забудь забрать детей в
239 00:20.";
240
241         TimeReducer timeReducer = new TimeReducer(message);
242         Console.WriteLine(timeReducer.ReduceTime(2));
243
244         string filePath = "words.bin";
245         string[] words = { "банан", "абрикос", "сок", "крекель",
246 "ламинат", "огурец" };
247
248         WordBinaryFile wordBinaryFile = new
249 WordBinaryFile(filePath);
250         wordBinaryFile.CreateFile(words);
251         wordBinaryFile.PrintWordsStartingWithLastLetter();
252
253         string inputFilePath = "input.txt";
254         string outputFilePath = "output.txt";
255
256         TextFileProcessor fileProcessor = new
257 TextFileProcessor(inputFilePath, outputFilePath);
258         fileProcessor.ProcessFile();
259
260         Console.WriteLine("Файл обработан. Результат сохранен в " +
261 outputFilePath);
262     }
263 }
264 }

```

**Вывод:** в результате работы мы получили навыки решения задач с применением ООП на языке C#.

### **Основная литература**

1. Зыков, С. В. Введение в теорию программирования. Объектно-ориентированный подход : учебное пособие / С. В. Зыков. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 187 с. — ISBN 978-5-4497-0926-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/102007.html>.
2. Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 245 с. — Текст : электронный — URL: <http://www.iprbookshop.ru/102051.html>.
3. Биллиг, В. А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) : учебник / В. А. Биллиг. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 409 с. — Текст : электронный — URL: <http://www.iprbookshop.ru/102029.html>.
4. Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т.I : учебник / С. В. Горелов ; под редакцией П. Б. Лукьянова. — Москва : Прометей, 2019. — 362 с. — Текст : электронный — URL: <http://www.iprbookshop.ru/94532.html>.
5. Горелов, С. В. Современные технологии программирования: разработка Windows-приложений на языке C#. В 2 томах. Т.II : учебник / С. В. Горелов ; под редакцией П. Б. Лукьянова. — Москва : Прометей, 2019. — 378 с. — Текст : электронный — URL: <http://www.iprbookshop.ru/94533.html>.