



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 11
по курсу «Численные методы линейной алгебры»
«Реализация мобильного приложения, выполняющего умножение
методом А.А.Карацубы»

Студент группы ИУ9-71Б Окутин Д.А.

Преподаватель Посевин Д. П.

Москва 2024

1 Задание

1. Реализовать метод А.А.Карацубы для умножения чисел.
2. Сравнить скорость работы с обычным умножением чисел.
3. Реализовать мобильное приложение, как точку входа в логику приложения

2 Реализация

Исходный код программы представлен в листинге 1.

Листинг 1: code

```
1 import 'dart:math';
2
3 import 'package:flutter/material.dart';
4
5 void main() {
6   runApp(MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       title: '...',
14       home: MultiplicationPage(),
15     );
16   }
17 }
18
19 class MultiplicationPage extends StatefulWidget {
20   @override
21   _MultiplicationPageState createState() => _MultiplicationPageState();
22 }
23
24 class _MultiplicationPageState extends State<MultiplicationPage> {
25   String number1 = '';
26   String number2 = '';
27   String binary1 = '';
28   String binary2 = '';
29   String result_kar = '';
30   String result_common = '';
31   String errorMessage = '';
32   String kar_t = '';
```

```

33 String comm_t = '';
34
35 void multiply() {
36     try {
37         int num1 = int.parse(number1);
38         int num2 = int.parse(number2);
39
40         //
41
42         if (num1.toRadixString(2).length % 2 != 0) {
43             throw Exception('
44
45                 ');
46         }
47
48         if (num2.toRadixString(2).length % 2 != 0) {
49             throw Exception('
50
51                 ');
52         }
53
54         //
55         Stopwatch stopwatch = new Stopwatch()..start();
56         result_kar = karatsuba(num1, num2).toString();
57         kar_t = stopwatch.elapsed.toString();
58         stopwatch = new Stopwatch()..start();
59         result_common = naiveMultiplication(num1,num2).toString();
60         comm_t = stopwatch.elapsed.toString();
61         setState(() {
62             binary1 = num1.toRadixString(2);
63             binary2 = num2.toRadixString(2);
64             errorMessage = '';
65         });
66     } catch (e) {
67         setState(() {
68             errorMessage = e.toString();
69             result_kar = '';
70         });
71     }
72 }
73
74 int karatsuba(int x, int y) {
75     if (x < (1 << 32) || y < (1 << 32)) {
76         return naiveMultiplication(x, y);
77     }
78 }

```

```

74     int n = (x.bitLength > y.bitLength ? x.bitLength : y.bitLength);
75     int half = n ~/ 2;
76
77     int highX = x >> half;
78     int lowX = x & ((1 << half) - 1);
79     int highY = y >> half;
80     int lowY = y & ((1 << half) - 1);
81
82     int z0 = karatsuba(lowX, lowY);
83     int z1 = karatsuba((lowX + highX), (lowY + highY));
84     int z2 = karatsuba(highX, highY);
85
86     return (z2 << (2 * half)) + ((z1 - z2 - z0) << half) + z0;
87 }
88
89 int naiveMultiplication(int x, int y) {
90     List<int> xDigits = x.toRadixString(2).split('').map(int.parse).
toList().reversed.toList();
91     List<int> yDigits = y.toRadixString(2).split('').map(int.parse).
toList().reversed.toList();
92
93     int result = 0;
94     for (int i = 0; i < yDigits.length; i++) {
95         int r = result;
96         for (int j = 0; j < xDigits.length; j++) {
97             result += (xDigits[j] * yDigits[i]) << (i + j);
98         }
99         result=r;
100        for (int j = 0; j < xDigits.length; j++) {
101            result += (xDigits[j] * yDigits[i]) << (i + j);
102        }
103    }
104
105    return result;
106 }
107
108 @override
109 Widget build(BuildContext context) {
110     return Scaffold(
111         appBar: AppBar(
112             title: Text(''),
113         ),
114         body: Padding(
115             padding: const EdgeInsets.all(16.0),
116             child: Column(
117                 children: [

```

```

118         TextField(
119             onChanged: (value) {
120                 number1 = value;
121             },
122             decoration: InputDecoration(labelText: '
123                                     '),
124             keyboardType: TextInputType.number,
125         ),
126         TextField(
127             onChanged: (value) {
128                 number2 = value;
129             },
130             decoration: InputDecoration(labelText: '
131                                     '),
132             keyboardType: TextInputType.number,
133         ),
134         SizedBox(height: 20),
135         ElevatedButton(
136             onPressed: multiply,
137             child: Text('
138                             '),
139         ),
140         SizedBox(height: 20),
141         if (errorMessage.isNotEmpty)
142             Text(errorMessage, style: TextStyle(color: Colors.red)),
143         if (binary1.isNotEmpty)
144             Text('
145                             : $binary1 ')
146     ,
147     if (binary2.isNotEmpty)
148         Text('
149                             : $binary2 ')
150     ,
151     SizedBox(height: 20),
152     if (result_kar.isNotEmpty)
153         Text('
154                             10-
155                             ): $result_kar\n
156                             : $kar_t ')
157     ,
158     if (result_common.isNotEmpty)
159         Text('
160                             10-
161                             ): $result_common \n
162                             :
163                             $comm_t '),
164     ],
165 ),
166 ),
167 );
168 }
169 }

```

3 Результаты

Результаты запуска представлены на рисунках 1.

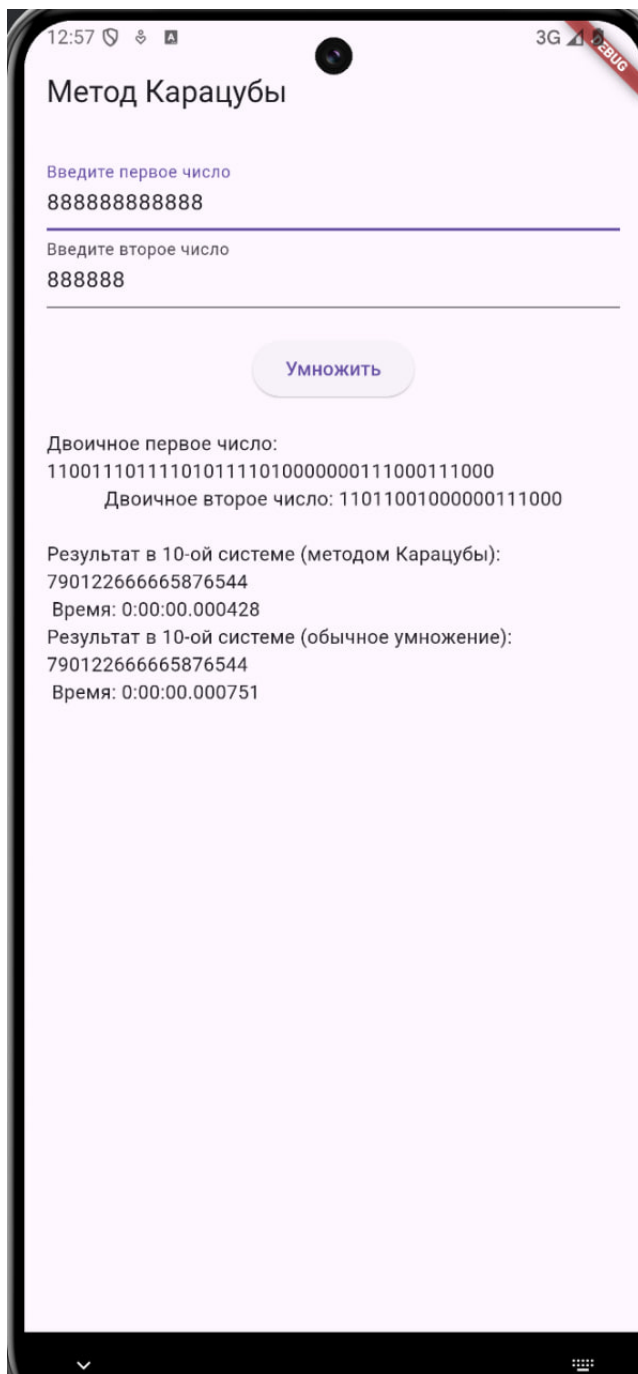


Рис. 1 — Результаты

4 Выводы

В результате выполнения данной лабораторной работы был реализован метод А.А.Карацубы для умножения чисел. Были проведены замеры времени для

демонстрации эффективности алгоритма. Результаты были оформлены в виде мобильного приложения.