



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 № 4**  
**по курсу «Разработка мобильных приложений»**  
**«Работа с AppBar и AnimationController»**

Студент группы ИУ9-71Б Окутин Д. А.

Преподаватель Посевин Д. П.

*Москва 2024*

# 1 Цель

Цель данной лабораторной работы: разобраться с использованием AppBar для создания меню приложения для маршрутизации по предыдущим лабораторным работам. Изучение AnimationController на примере создания анимированного графика параболы.

## 2 Задание

1. Создать приложение с AppBar виджетом.
2. Добавить всплывающее меню с возможностью открытия предыдущих лабораторных работ.
3. Использовать AnimationController для анимированного графика параболы на главной странице приложения.

## 3 Реализация

Исходный код представлен в листинге 1.

Листинг 1: Исхоный код программы

```
1
2   import 'package:flutter/material.dart';
3 import 'package:http/http.dart' as http;
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       home: HomeScreen(), // Scaffold
14       debugShowCheckedModeBanner: false,
15     );
16   }
17 }
18
19 class HomeScreen extends StatelessWidget {
```

```

20  @override
21  Widget build(BuildContext context) {
22    return Scaffold(
23      appBar: AppBar(
24        title: const Text("          4"),
25        backgroundColor: Colors.greenAccent[400],
26        elevation: 50.0,
27      ), //AppBar
28      drawer: Drawer(
29        child: ListView(
30          padding: EdgeInsets.zero,
31          children: <Widget>[
32            Container(
33              height: 150.0,
34              color: Colors.greenAccent[400],
35              child: Center(
36                child: Text(
37                  '          ',
38                  style: TextStyle(
39                    color: Colors.black,
40                    fontSize: 24,
41                  ),
42                ),
43              ),
44            ),
45            ListTile(
46              title: Text('          2'),
47              onTap: () {
48                Navigator.push(
49                  context,
50                  MaterialPageRoute(builder: (context) => Lab2(title: '
Lab2',)),
51                );
52              },
53            ),
54            ListTile(
55              title: Text('          3'),
56              onTap: () {
57                Navigator.push(
58                  context,
59                  MaterialPageRoute(builder: (context) => Lab3(title: '
Lab3',)),
60                );
61              },
62            ),
63          ],

```

```

64         ),
65     ),
66     body: Center(
67         child: Lab4(),
68     ), // Center
69 );
70 }
71 }
72
73 class Lab4 extends StatelessWidget {
74     @override
75     Widget build(BuildContext context) {
76         return MaterialApp(
77             home: ParabolaPage(),
78             debugShowCheckedModeBanner: false,
79         );
80     }
81 }
82
83 class ParabolaPage extends StatefulWidget {
84     @override
85     _ParabolaPageState createState() => _ParabolaPageState();
86 }
87
88 class _ParabolaPageState extends State<ParabolaPage>
89     with SingleTickerProviderStateMixin {
90     late AnimationController _controller;
91     late Animation<double> _animation;
92
93     double a = 1.0;
94     double b = 0.0;
95     double c = 0.0;
96
97     @override
98     void initState() {
99         super.initState();
100         _controller = AnimationController(
101             duration: const Duration(seconds: 40),
102             vsync: this,
103         )..repeat(reverse: true);
104
105         _animation = Tween<double>(begin: -15.0, end: 15.0).animate(
106             _controller);
107
108     @override

```

```

109 void dispose() {
110     _controller.dispose();
111     super.dispose();
112 }
113
114 @override
115 Widget build(BuildContext context) {
116     return Scaffold(
117         body: Column(
118             children: [
119                 Expanded(
120                     child: Center(
121                         child: AnimatedBuilder(
122                             animation: _animation,
123                             builder: (context, child) {
124                                 return CustomPaint(
125                                     size: Size(double.infinity, double.infinity),
126                                     painter: ParabolaPainter(_animation.value, b, c),
127                                 );
128                             },
129                         ),
130                     ),
131                 ),
132                 _buildSlider("a", -100.0, 100.0, (value) {
133                     setState(() {
134                         a = value;
135                     });
136                 }),
137                 _buildSlider("b", -100.0, 100.0, (value) {
138                     setState(() {
139                         b = value;
140                     });
141                 }),
142                 _buildSlider("c", -100.0, 100.0, (value) {
143                     setState(() {
144                         c = value;
145                     });
146                 }),
147             ],
148         ),
149     );
150 }
151
152 Widget _buildSlider(String label, double min, double max, ValueChanged
153     <double> onChanged) {
    return Column(

```

```

154     children: [
155         Text(label),
156         Slider(
157             value: label == "a" ? a : label == "b" ? b : c,
158             min: min,
159             max: max,
160             onChanged: onChanged,
161             divisions: 200,
162             label: label == "a" ? a.toStringAsFixed(2) : label == "b" ? b.
toStringAsFixed(2) : c.toStringAsFixed(2),
163         ),
164     ],
165 );
166 }
167 }
168
169 class ParabolaPainter extends CustomPainter {
170     final double a;
171     final double b;
172     final double c;
173
174     ParabolaPainter(this.a, this.b, this.c);
175
176     @override
177     void paint(Canvas canvas, Size size) {
178         final paint = Paint()
179             ..color = Colors.blue
180             ..style = PaintingStyle.stroke
181             ..strokeWidth = 2;
182
183         final path = Path();
184
185         //
186         for (double x = -size.width / 2; x <= size.width / 2; x += 0.1) {
187             double y = a * x * x / 100 + b * x + c; //
188                 y = ax^2 + bx + c
189
190             if (x == -size.width / 2) {
191                 path.moveTo(size.width / 2 + x, size.height / 2 - y);
192             } else {
193                 path.lineTo(size.width / 2 + x, size.height / 2 - y);
194             }
195         }
196
197         canvas.drawPath(path, paint);
198     }
199 }

```

```

198
199     @override
200     bool shouldRepaint(ParabolaPainter oldDelegate) {
201         return true;
202     }
203 }
204
205 class Lab2 extends StatefulWidget {
206     const Lab2({Key? key, required this.title}) : super(key: key);
207     final String title;
208
209     @override
210     State<Lab2> createState() => _MyHomePageState();
211 }
212
213 class _MyHomePageState extends State<Lab2> {
214     int _counter = 0;
215
216     void _incrementCounter() {
217         setState(() {
218             _counter++;
219         });
220     }
221
222     void _getRequestOn() {
223         setState(() {
224             http.get(Uri.parse("http://iocontrol.ru/api/sendData/lab1_panel/
lab1/1")).then((response) {
225                 print("Response status: ${response.statusCode}");
226                 print("Response body: ${response.body}");
227             }).catchError((error){
228                 print("Error: $error");
229             });
230         });
231     }
232
233     void _getRequestOff() {
234         setState(() {
235             http.get(Uri.parse("http://iocontrol.ru/api/sendData/lab1_panel/
lab1/0")).then((response) {
236                 print("Response status: ${response.statusCode}");
237                 print("Response body: ${response.body}");
238             }).catchError((error){
239                 print("Error: $error");
240             });
241         });

```

```

242 }
243
244 @override
245 Widget build(BuildContext context) {
246   return Scaffold(
247     appBar: AppBar(
248       title: Text(widget.title),
249     ),
250     body: Center(
251       child: Column(
252         mainAxisAlignment: MainAxisAlignment.center,
253         children: <Widget>[
254           const Text(
255             'You have pushed the button this many times:',
256           ),
257           Text(
258             '$_counter',
259             style: Theme.of(context).textTheme.headlineLarge,
260           ),
261
262           TextButton(
263             style: ButtonStyle(
264               foregroundColor: MaterialStateProperty.all<Color>(Colors
265                 .blue),
266             ),
267             onPressed: _getRequestOn,
268             child: Text('On'),
269           ),
270
271           TextButton(
272             style: ButtonStyle(
273               foregroundColor: MaterialStateProperty.all<Color>(Colors
274                 .blue),
275             ),
276             onPressed: _getRequestOff,
277             child: Text('Off'),
278           ),
279         ],
280       ),
281       floatingActionButton: FloatingActionButton(
282         onPressed: _incrementCounter,
283         tooltip: 'Increment',
284         child: const Icon(Icons.add),
285       ),

```



```

286
287     );
288 }
289 }
290
291 class Lab3 extends StatefulWidget {
292     const Lab3({Key? key, required this.title}) : super(key: key);
293     final String title;
294
295     @override
296     State<Lab3> createState() => _NumberFormState();
297 }
298
299 class _NumberFormState extends State<Lab3> {
300     final _formKey = GlobalKey<FormState>();
301     final _numberController = TextEditingController();
302     int _currentNumber = 0;
303
304     Future<void> sendNumber() async {
305         final url = Uri.parse('http://195.19.55.124:8080/');
306         final response = await http.post(
307             url,
308             headers: {
309                 'Content-Type': 'text/plain',
310             },
311             body: '$_currentNumber',
312         );
313
314         if (response.statusCode == 200) {
315             print('Num sent: ${_currentNumber}');
316         } else {
317             print('Error: ${response.statusCode}');
318         }
319     }
320
321     Future<void> sendNumberFromNumberController() async {
322         final url = Uri.parse('http://195.19.55.124:8080/');
323         final response = await http.post(
324             url,
325             headers: {
326                 'Content-Type': 'text/plain',
327             },
328             body: _numberController.text,
329         );
330
331         if (response.statusCode == 200) {
332             print('Num sent: ${_numberController.text}');

```

```

332     } else {
333         print('Error: ${response.statusCode}');
334     }
335 }
336
337 Future<void> getNumber() async {
338     final url = Uri.parse('http://195.19.55.124:8080/');
339
340     final response = await http.get(url);
341
342     if (response.statusCode == 200) {
343         setState(() {
344             _currentNumber = int.parse(response.body);
345         });
346
347         print('Num: $_currentNumber');
348     } else {
349         print('Error: ${response.statusCode}');
350     }
351 }
352
353 void increment() {
354     setState(() {
355         _currentNumber++;
356     });
357
358     sendNumber();
359 }
360
361 void decrement() {
362     setState(() {
363         _currentNumber--;
364     });
365
366     sendNumber();
367 }
368
369 @override
370 Widget build(BuildContext context) {
371     return Scaffold(
372         appBar: AppBar(
373             title: Text('Lab3'),
374         ),
375         body: Padding(
376             padding: const EdgeInsets.all(16.0),
377             child: Form(

```

```

378         key: _formKey,
379         child: Column(
380           children: <Widget>[
381             TextFormField(
382               controller: _numberController,
383               decoration: InputDecoration(labelText: '
384             ),
385             keyboardType: TextInputType.number,
386             validator: (value) {
387               if (value == null || value.isEmpty) {
388                 return '
389               }
390               return null;
391             },
392             const SizedBox(height: 20),
393             ElevatedButton(
394               onPressed: () {
395                 if (_formKey.currentState!.validate()) {
396                   sendNumberFromNumberController();
397                 }
398               },
399               child: Text('
400             ),
401             SizedBox(height: 20),
402             ElevatedButton(
403               onPressed: getNumber,
404               child: Text('
405             ),
406             const SizedBox(height: 20),
407             Text('
408           TextStyle(fontSize: 20)),
409             SizedBox(height: 20),
410             Row(
411               mainAxisAlignment: MainAxisAlignment.center,
412               children: [
413                 ElevatedButton(
414                   onPressed: decrement,
415                   child: Text('
416                 ),
417                 SizedBox(width: 20),
418                 ElevatedButton(
419                   onPressed: increment,
420                   child: Text('
421               ],

```

```
422         ),
423         ],
424     ),
425     ),
426     ),
427 );
428 }
429 }
```

## 4 Результаты

Результаты представлено на рисунках 1 - 2.



Рис. 1 — Вид главной страницы приложения



Рис. 2 — Вид страницы с меню

## **5 Выводы**

В результате данной лабораторной работы были изучены методологии работы с AppBar и AnimationController. Успешно соединены наработки прошлых лабораторных работ в одном мобильном приложении.