Міністерство освіти і науки України
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЗВІТ

з лабораторної роботи №3.1 з дисципліни
«Інтелектуальні вбудовані системи»

Виконав:
ІП-83
Сергійчук Н. С.

Київ 2020

1. Завдання

2. Розробити програма для факторизації заданого числа методом Ферма. Реалізувати користувацький інтерфейс з можливістю вводу даних.

3. Лістинг програми

```
// Binary searched square root
import 'dart:math';


int intSqrt(int x) {
var a = 0;
var b = x;
while (b - a > 1) {
var mid = (a + b) ~/ 2;
var sqr = mid * mid;
if (sqr == x) {
return mid;
} else if (sqr < x) {
a = mid;
} else {
b = mid;
}
}
return a;
}
```

```
bool isSquare(int x) {
var sqrt = intSqrt(x);
return sqrt * sqrt == x;
}
```

```
// x should be odd
List<int> fermat(int n) {
assert(n.isOdd);
int a = sqrt(n).ceil();
int b2 = a * a - n;
int b = sqrt(b2).round();
while (b * b != b2) {
a = a + 1;
b2 = a * a - n;
b = sqrt(b2).round();
}
return [a - b, a + b];
}
```

```
List<int> factorInner(int x) {
assert(x > 0);
if (x <= 3) {
return [x];
}
```

```
var result = <int>[];
while (x.isEven) {
x ~/= 2;
result.add(2);
}
var factors = fermat(x);
for (var f in factors) {
```

```
if (f == x) {
result.add(x);
} else {
result.addAll(fermat(f));
}
}

return result;
}

List<int> factor(int x) {
var primes = factorInner(x);
var withoutone = primes.where((element) => element != 1).toList();
withoutone.sort();
return withoutone;
}

int randBetwen(Random rng, int a, int b) {
var num = rng.nextInt(b - a);
return num + a;
}

int testDuration(order, int tests) {
var rng = Random();

var delta = max((order * 0.1).round() as int, 1.0).round();
var sum = 0;
for (var i = 0; i < tests; i++) {
```

```
var number = order + randBetwen(rng, -delta, delta);
if (number < 0) {
number = 2;
}
Stopwatch stopwatch = new Stopwatch()..start();
factor(number);
sum += stopwatch.elapsed.inMicroseconds;
}
```

```
sum ~/= tests;
return sum;
}
```

```
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:charts_flutter/flutter.dart' as charts;

import 'ferma.dart';
```

```
void main() {
runApp(MyApp());
}
```

```
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
```

```
return MaterialApp(
title: 'Flutter Demo',
theme: ThemeData(
primarySwatch: Colors.blue,
),
home: MyHomePage(title: 'Ferma Factorization'),
);
}
}
```

```
class MyHomePage extends StatefulWidget {
MyHomePage({Key key, this.title}) : super(key: key);
```

```
final String title;
```

```
@override
_MyHomePageState createState() => _MyHomePageState();
}
```

```
class _MyHomePageState extends State<MyHomePage> {
String number = "";
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: Text(widget.title),
),
```

```
body: Container(
padding: const EdgeInsets.all(40.0),
child: Center(
child: Column(
mainAxisAlignment: MainAxisAlignment.start,
children: <Widget>[
new TextField(
decoration: new InputDecoration(labelText: "Enter your number"),
keyboardType: TextInputType.number,
inputFormatters: <TextInputFormatter>[
FilteringTextInputFormatter.digitsOnly
],
onChanged: (String value) => _setNumber(value),
),
Expanded(
child: Padding(
padding: EdgeInsets.only(top: 40.0, bottom: 40.0),
child: _buildFactors(),
)),
],
),
),
),
floatingActionButton: FloatingActionButton(
tooltip: 'Measures',
child: Icon(Icons.show_chart_sharp),
onPressed: () => {
Navigator.push(context,
```

```
MaterialPageRoute(builder: (context) => ChartPage.init()))
},
), // This trailing comma makes auto-formatting nicer for build methods.
);
}
```

```
void _setNumber(String num) {
setState(() {
number = num;
});
}
```

```
List<int> _getFactors(String num) {
var n = int.tryParse(num);
if (n == null || n < 0) {
return null;
}
return factor(n);
}
```

```
Widget _buildFactors() {
var factors = _getFactors(number);
if (factors == null) {
return Text("Unsupported number");
}
return ListView.separated(
padding: const EdgeInsets.all(8),
itemCount: factors.length,
```

```
itemBuilder: (BuildContext context, int index) {
return Container(
height: 50,
child: Text('${factors[index]}'),
);
},
separatorBuilder: (BuildContext context, int index) => const Divider(),
);
}
}
```

```
class ChartPage extends StatelessWidget {
final List<Point> measures;
```

```
ChartPage(this.measures);
```

```
factory ChartPage.init() {
var points = <Point>[];
for (int i = 16; i < 2 << 17; i *= 2) {
print("Processing $i");
points.add(Point(i, testDuration(i, 100)));
}
return ChartPage(points);
}
```

```
@override
Widget build(BuildContext context) {
return Scaffold(
```

```
appBar: AppBar(
title: Text('Measures'),
),
body: Chart(measures));
}
}
```

```
class Chart extends StatelessWidget {
final List<Point> points;
```

```
Chart(this.points);
```

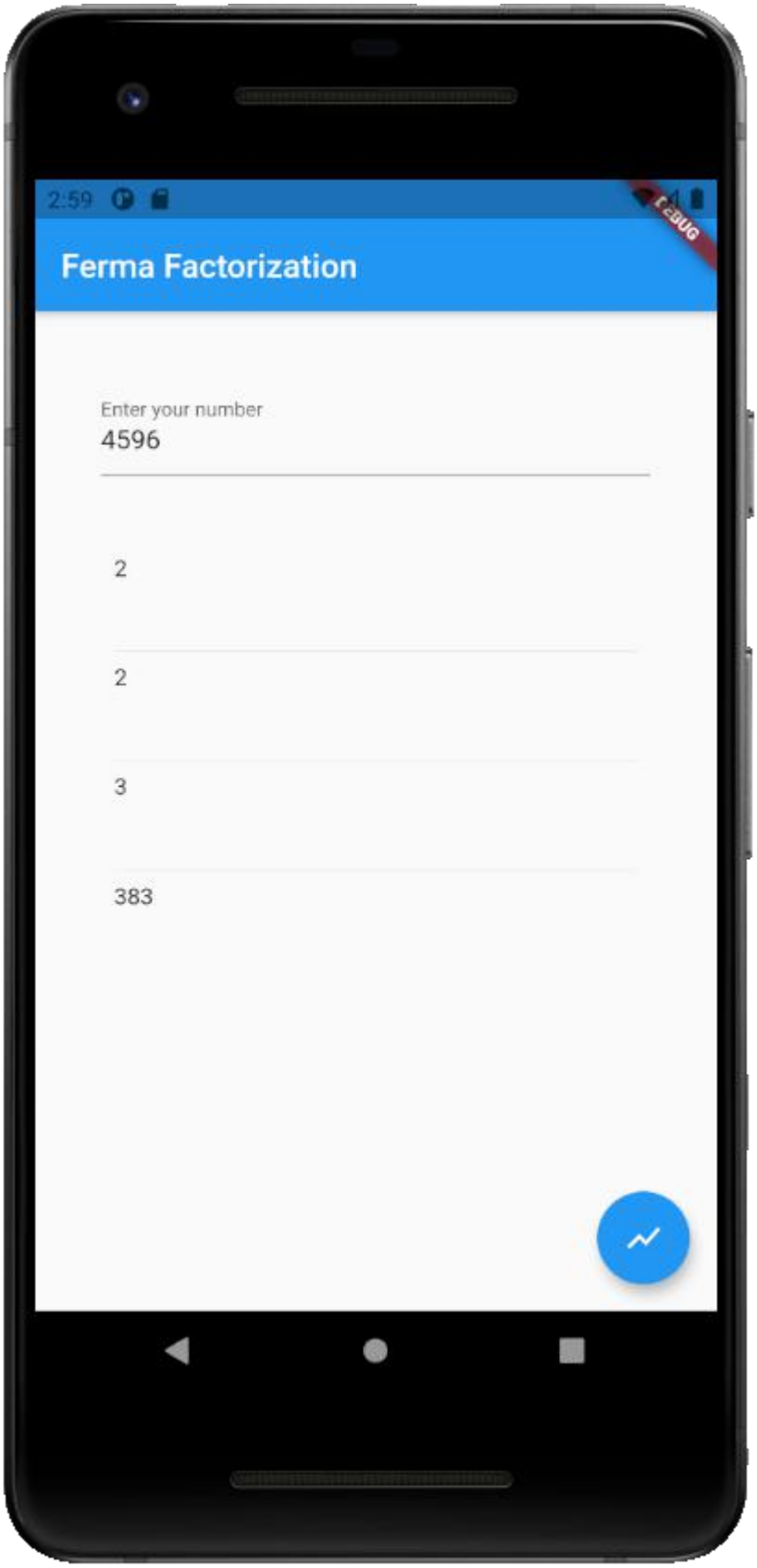
```
@override
Widget build(BuildContext context) {
var serial = charts.Series<Point, int>(
id: 'points',
colorFn: (_, __) => charts.MaterialPalette.blue.shadeDefault,
domainFn: (Point p, _) => p.x,
measureFn: (Point p, _) => p.y,
data: points,
);
var series = [serial];
return new charts.LineChart(series, animate: false);
}
}
```

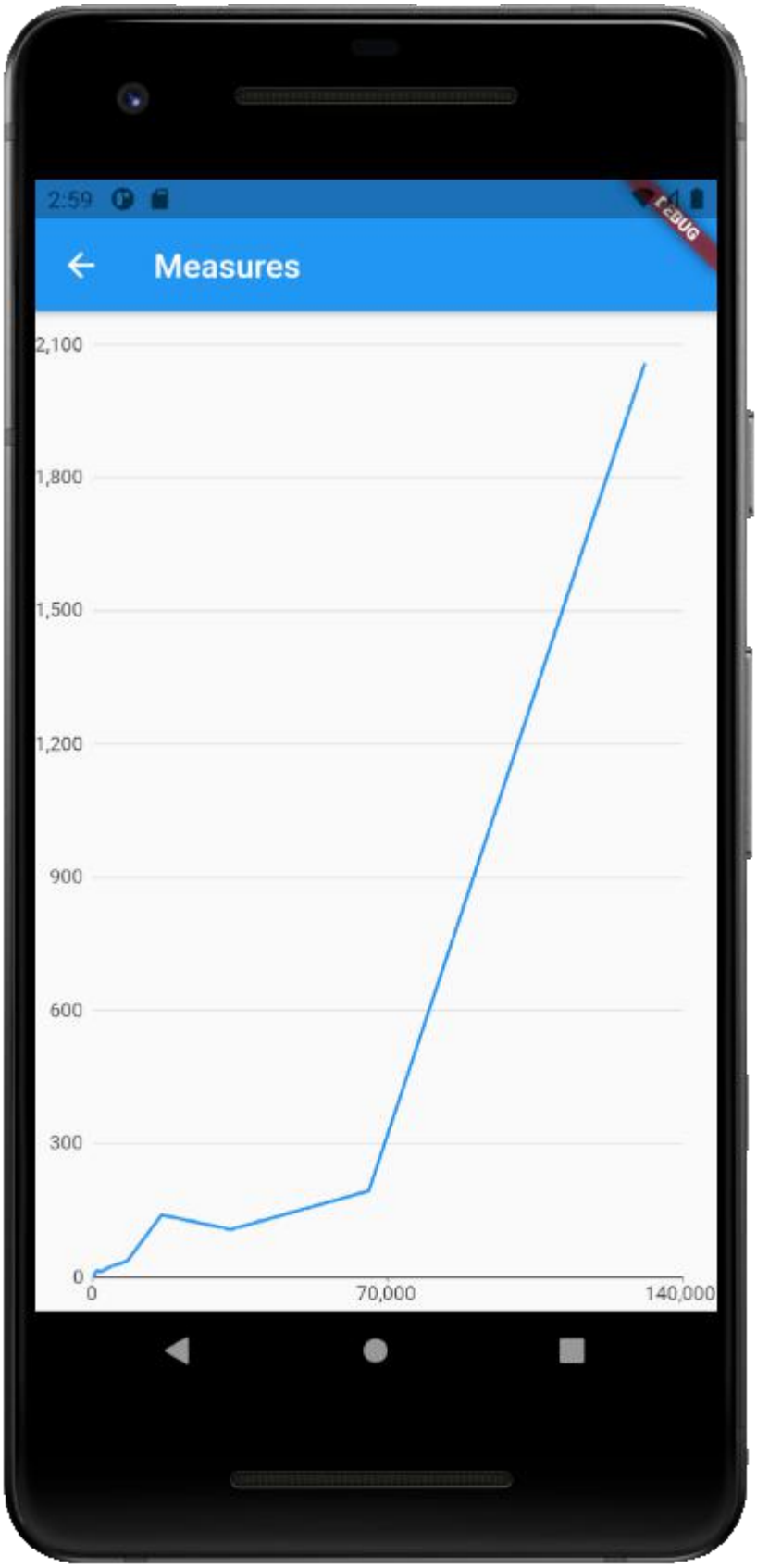
```
class Point {
int x;
```

```
int y;
```

```
Point(this.x, this.y);
}
```

4. Результати виконання кожної програми.


2:59
Debug
Ferma Factorization
Enter your number
4596
2
2
3
383


2:59
Measures
2,100
1,800
1,500
1,200
900
600
300
0
0
70,000
140,000