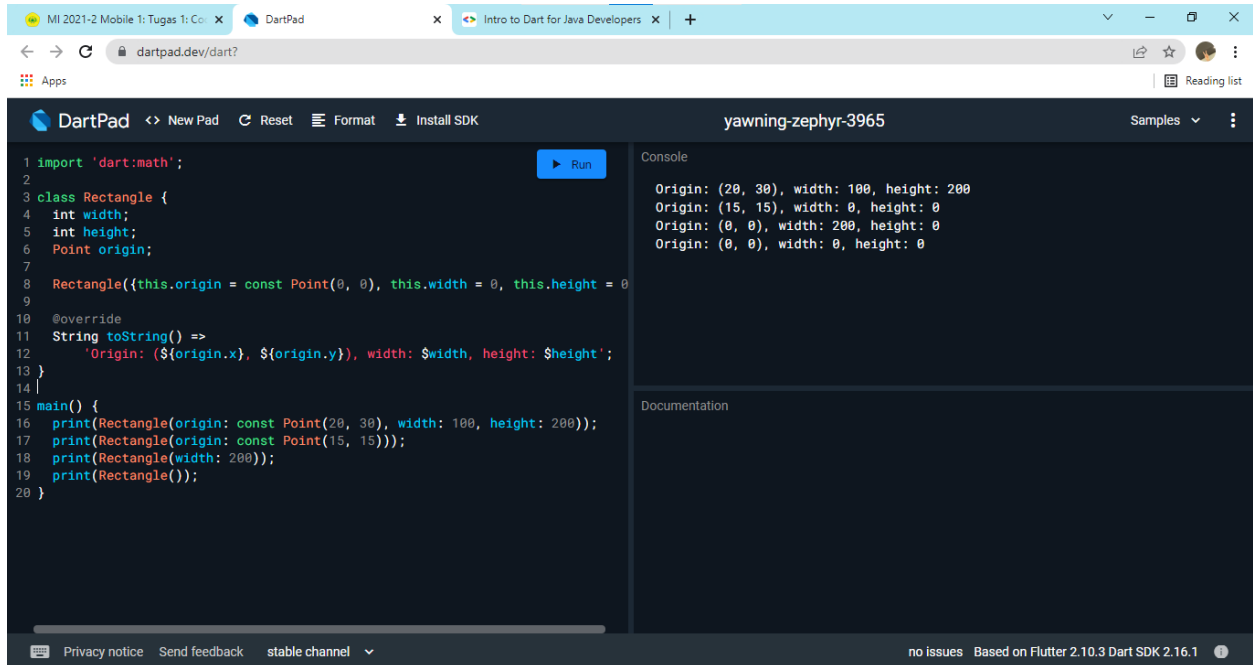


Nama: Revi Ruvina Ratu

NPM: 085020021

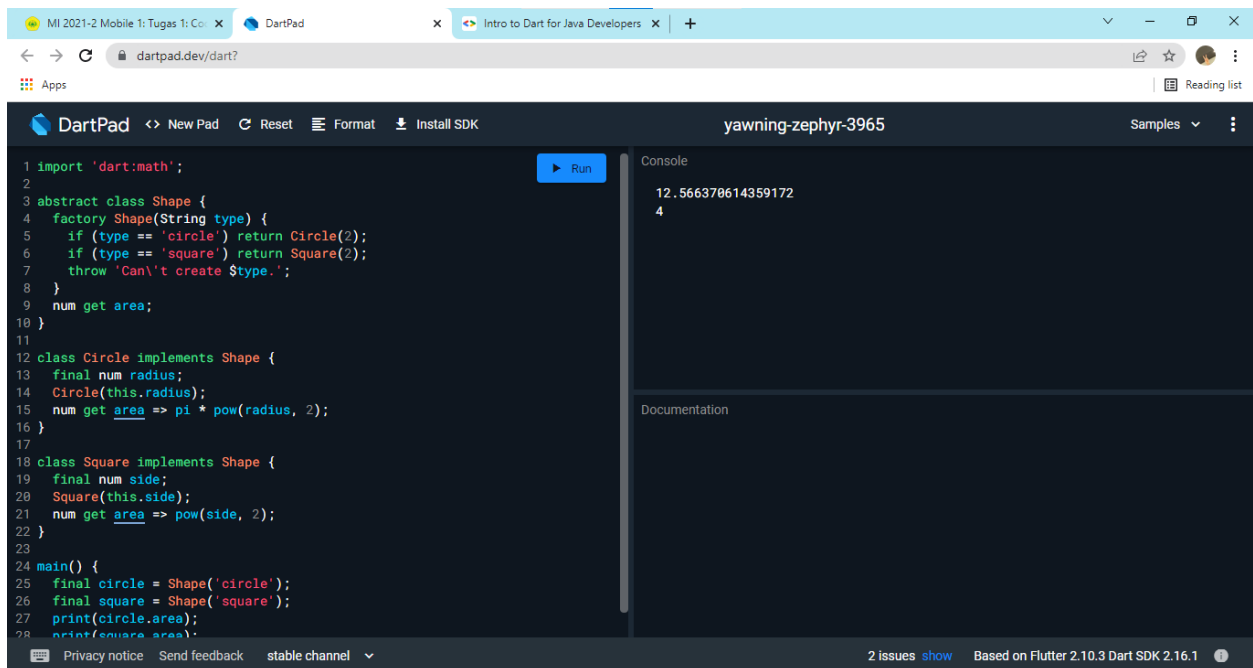


The screenshot shows the DartPad web interface. The code editor contains a Dart class named `Rectangle` with attributes `width`, `height`, and `origin` (a `Point` object). The `toString` method is overridden to return a string representation of the rectangle. The `main` function creates and prints three `Rectangle` objects.

```
1 import 'dart:math';
2
3 class Rectangle {
4   int width;
5   int height;
6   Point origin;
7
8   Rectangle({this.origin = const Point(0, 0), this.width = 0, this.height = 0})
9
10  @override
11  String toString() =>
12    'Origin: (${origin.x}, ${origin.y}), width: $width, height: $height';
13 }
14
15 main() {
16   print(Rectangle(origin: const Point(20, 30), width: 100, height: 200));
17   print(Rectangle(origin: const Point(15, 15)));
18   print(Rectangle(width: 200));
19   print(Rectangle());
20 }
```

The console output shows the string representation of the rectangles:

```
Origin: (20, 30), width: 100, height: 200
Origin: (15, 15), width: 0, height: 0
Origin: (0, 0), width: 200, height: 0
Origin: (0, 0), width: 0, height: 0
```



The screenshot shows the DartPad web interface. The code editor contains a Dart class hierarchy for shapes. An abstract `Shape` class has a `factory` constructor and an `area` property. `Circle` and `Square` classes implement `Shape` and override the `area` property.

```
1 import 'dart:math';
2
3 abstract class Shape {
4   factory Shape(String type) {
5     if (type == 'circle') return Circle(2);
6     if (type == 'square') return Square(2);
7     throw "Can't create $type.";
8   }
9   num get area;
10 }
11
12 class Circle implements Shape {
13   final num radius;
14   Circle(this.radius);
15   num get area => pi * pow(radius, 2);
16 }
17
18 class Square implements Shape {
19   final num side;
20   Square(this.side);
21   num get area => pow(side, 2);
22 }
23
24 main() {
25   final circle = Shape('circle');
26   final square = Shape('square');
27   print(circle.area);
28   print(square.area);
29 }
```

The console output shows the area of the shapes:

```
12.566370614359172
4
```

MI 2021-2 Mobile 1: Tugas 1: Co... DartPad Intro to Dart for Java Developers +

dartpad.dev/dart?

Apps Reading list

DartPad <> New Pad Reset Format Install SDK yawning-zephyr-3965 Samples

```
1 import 'dart:math';
2
3 abstract class Shape {
4   factory Shape(String type) {
5     if (type == 'circle') return Circle(2);
6     if (type == 'square') return Square(2);
7     throw 'Can't create $type.';
8   }
9   num get area;
10 }
11
12 class Circle implements Shape {
13   final num radius;
14   Circle(this.radius);
15   num get area => pi * pow(radius, 2);
16 }
17
18 class Square implements Shape {
19   final num side;
20   Square(this.side);
21   num get area => pow(side, 2);
22 }
23
24 class CircleMock implements Circle {
25   num area = 0;
26   num radius = 0;
27 }
28
```

Run

Console

```
12.566370614359172
4
```

Documentation

num radius

Privacy notice Send feedback stable channel 4 issues show Based on Flutter 2.10.3 Dart SDK 2.16.1

MI 2021-2 Mobile 1: Tugas 1: Co... DartPad Intro to Dart for Java Developers +

dartpad.dev/dart?

Apps Reading list

DartPad <> New Pad Reset Format Install SDK yawning-zephyr-3965 Samples

```
1 String scream(int length) => "A${'a' * length}h!";
2
3 main() {
4   final values = [6, 7, 8, 5, 10, 50];
5   values.skip(1).take(3).map(scream).forEach(print);
6 }
```

Run

Console

```
Aaaaaaaah!
Aaaaaaaah!
Aaaaaah!
```

Documentation

Privacy notice Send feedback stable channel no issues Based on Flutter 2.10.3 Dart SDK 2.16.1