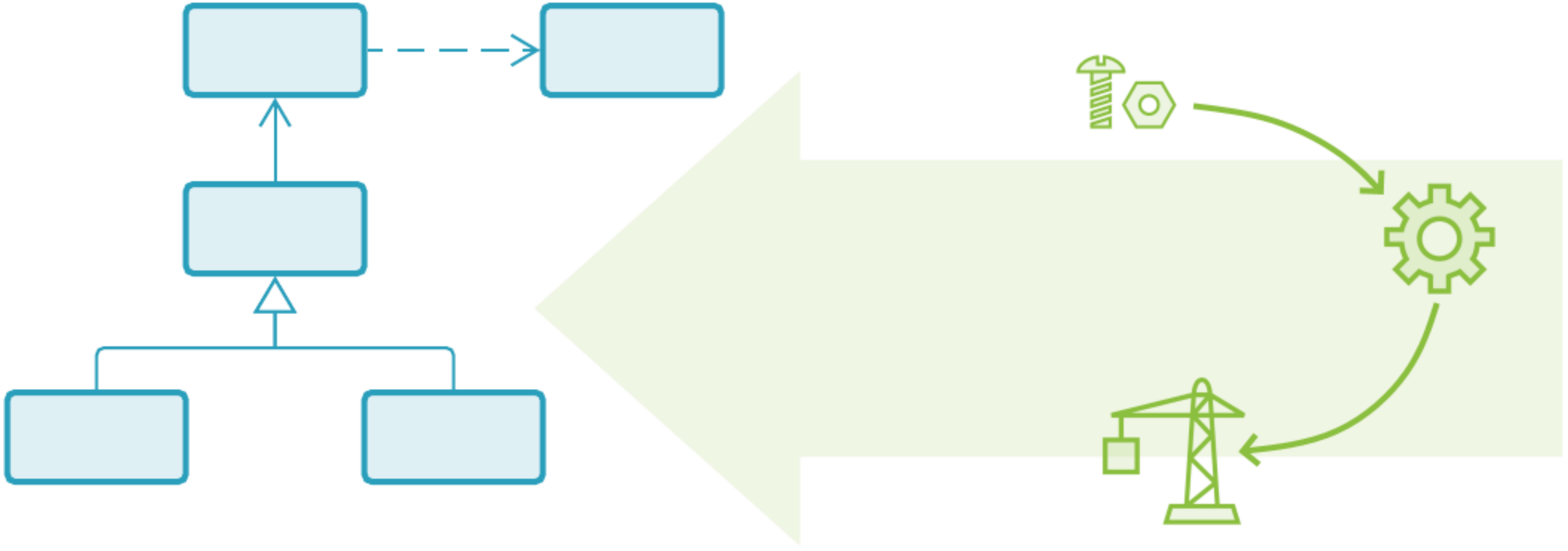
Advancing to a Domain-specific Language

Zoran Horvat
CEO AT CODING HELMET

@zoranh75 http://codinghelmet.com

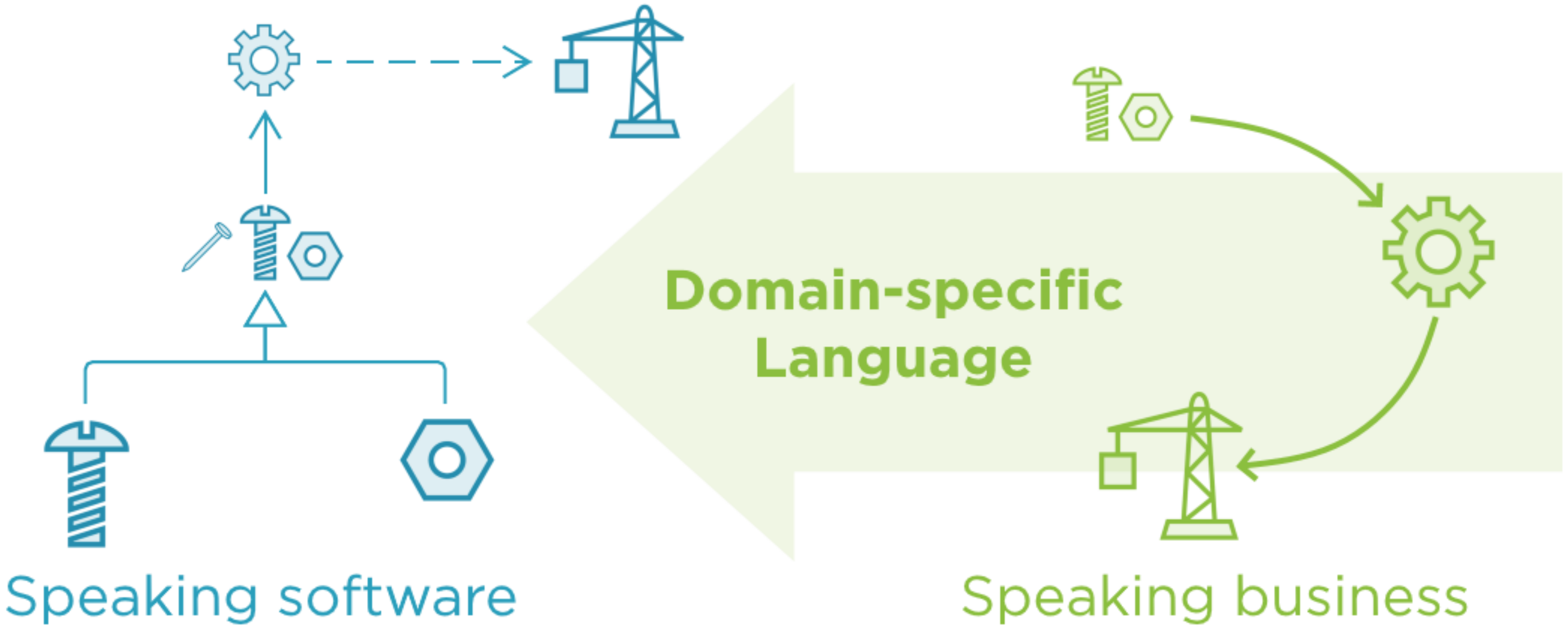
Unifying the Languages



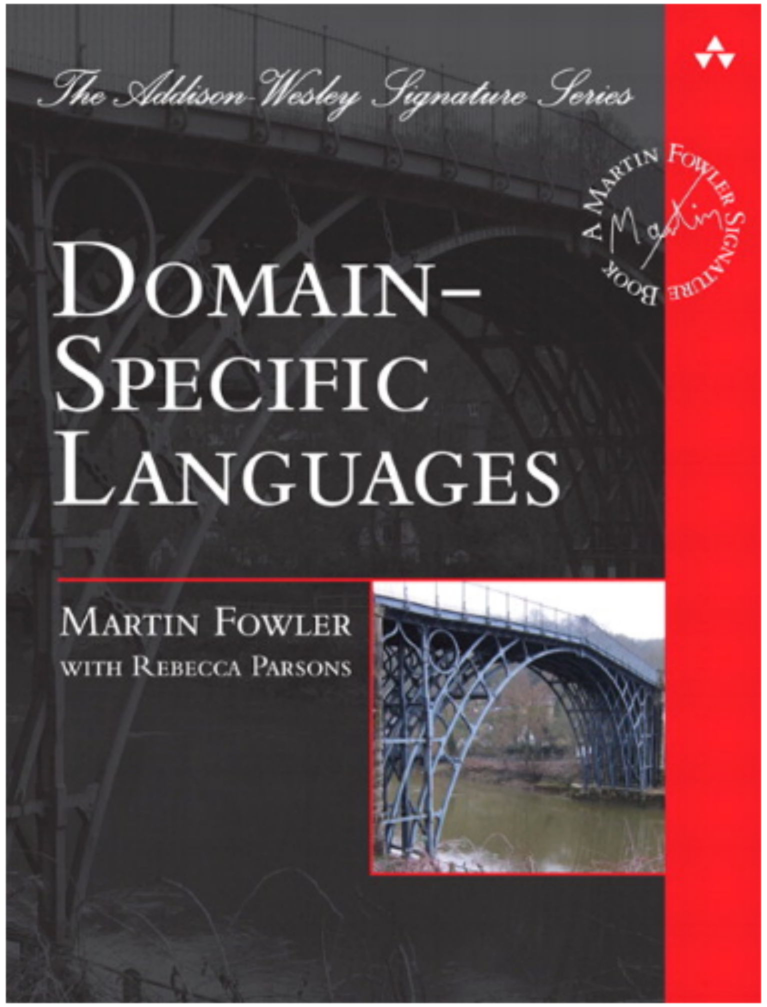
Speaking software

Speaking business

Unifying the Languages



Introducing Domain-specific Languages


The Addison-Wesley Signature Series
A Martin Fowler Signature Book
Domain-Specific Languages
Martin Fowler
with Rebecca Parsons

Martin Fowler

Domain-specific Languages

(September 2010)

Introducing Domain-specific Languages

Compose elements of code the same way we compose sentences in a spoken language

```
painters
  .thoseAvailable()
  .findFastestOne()
  .paint(area)
```

Introducing Domain-specific Languages

Compose elements of code the same way we compose sentences in a spoken language

```
painters
  .available()
  .fastest()
  .paint(area)
  .orElse(report::noWorkDone)
```

Internal DSL

Introducing Domain-specific Languages

Compose elements of code the same way we compose sentences in a spoken language

```
painters
   .available()
   .fastest()
   .paint(area)
   .orElse(report::noWorkDone)
```

Internal DSL

```
take available @painters
find fastest one
let it paint @area
or else report no work done
```

External DSL

Introducing Domain-specific Languages

Compose elements of code the same way we compose sentences in a spoken language

```
painters
   .available()
   .fastest()
   .paint(area)
   .orElse(report::noWorkDone)
```

```
take available @painters
find fastest one
let it paint @area
or else report no work done
```

Introducing Domain-specific Languages

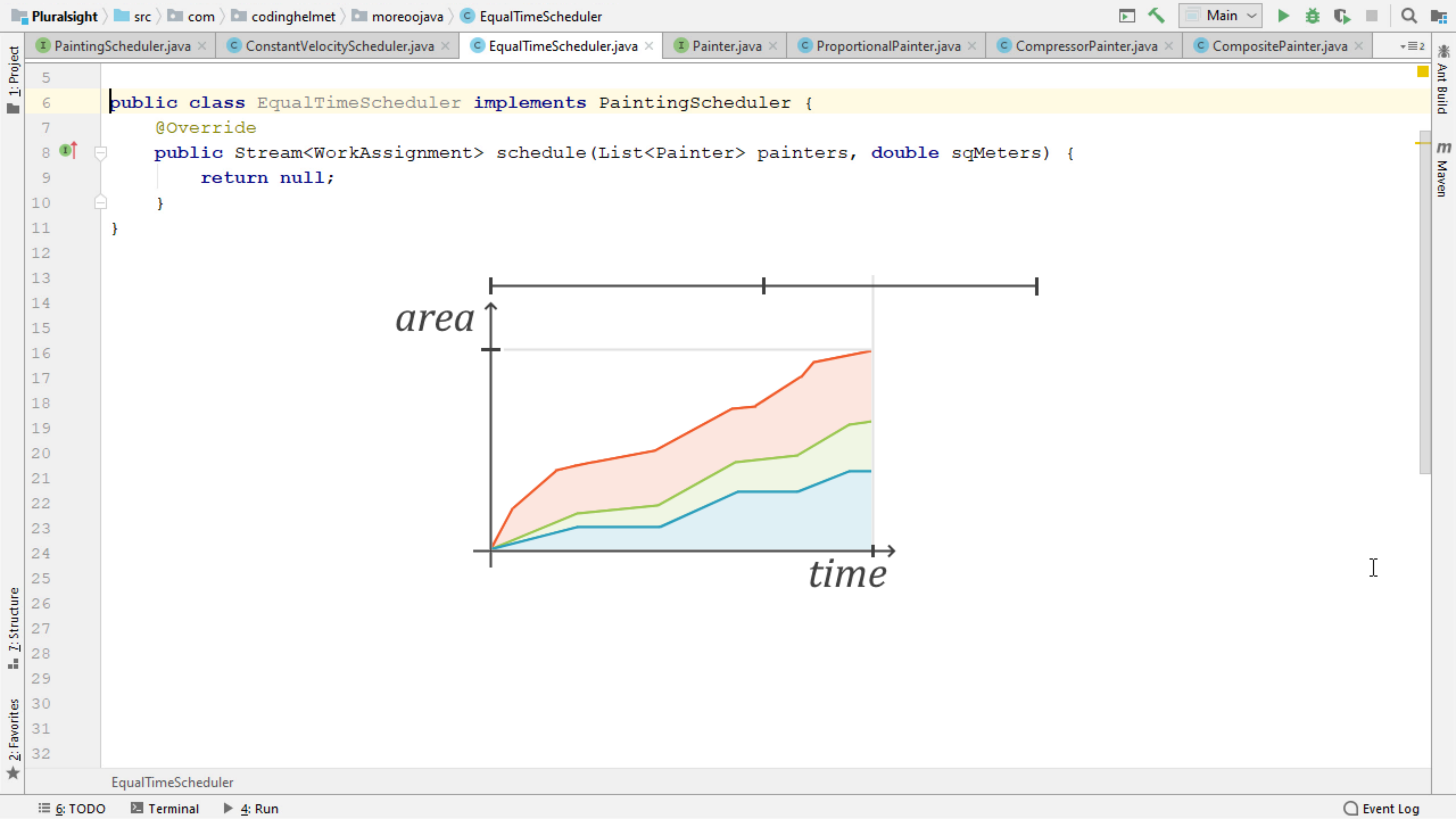
Compose elements of code the same way we compose sentences in a spoken language

```
painters
   .available()
   .fastest()
   .paint(area)
   .orElse(report::noWorkDone)
```

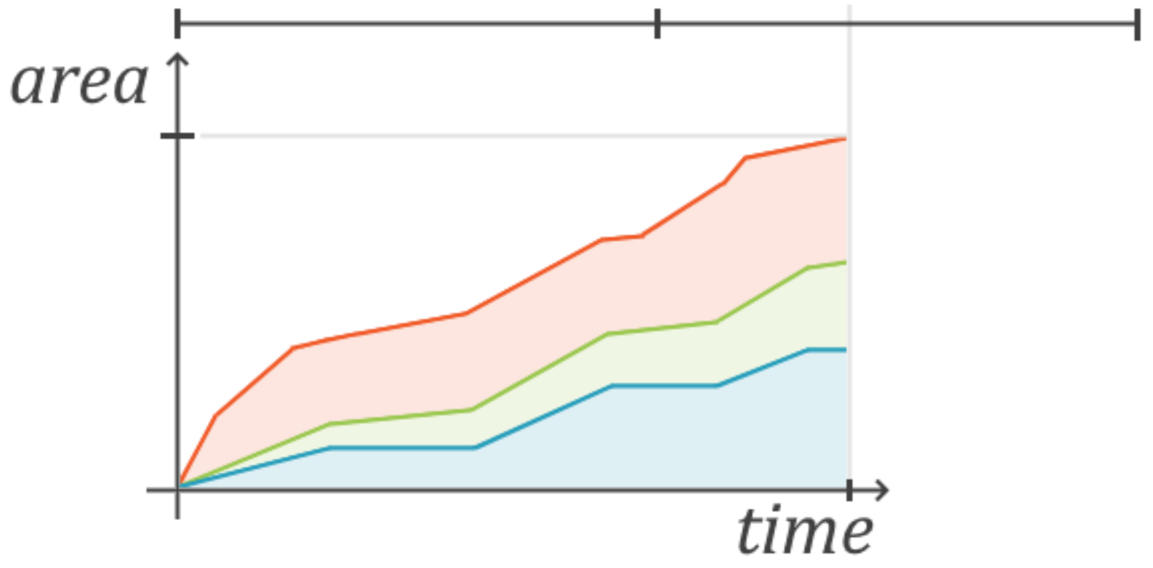
Used by developers

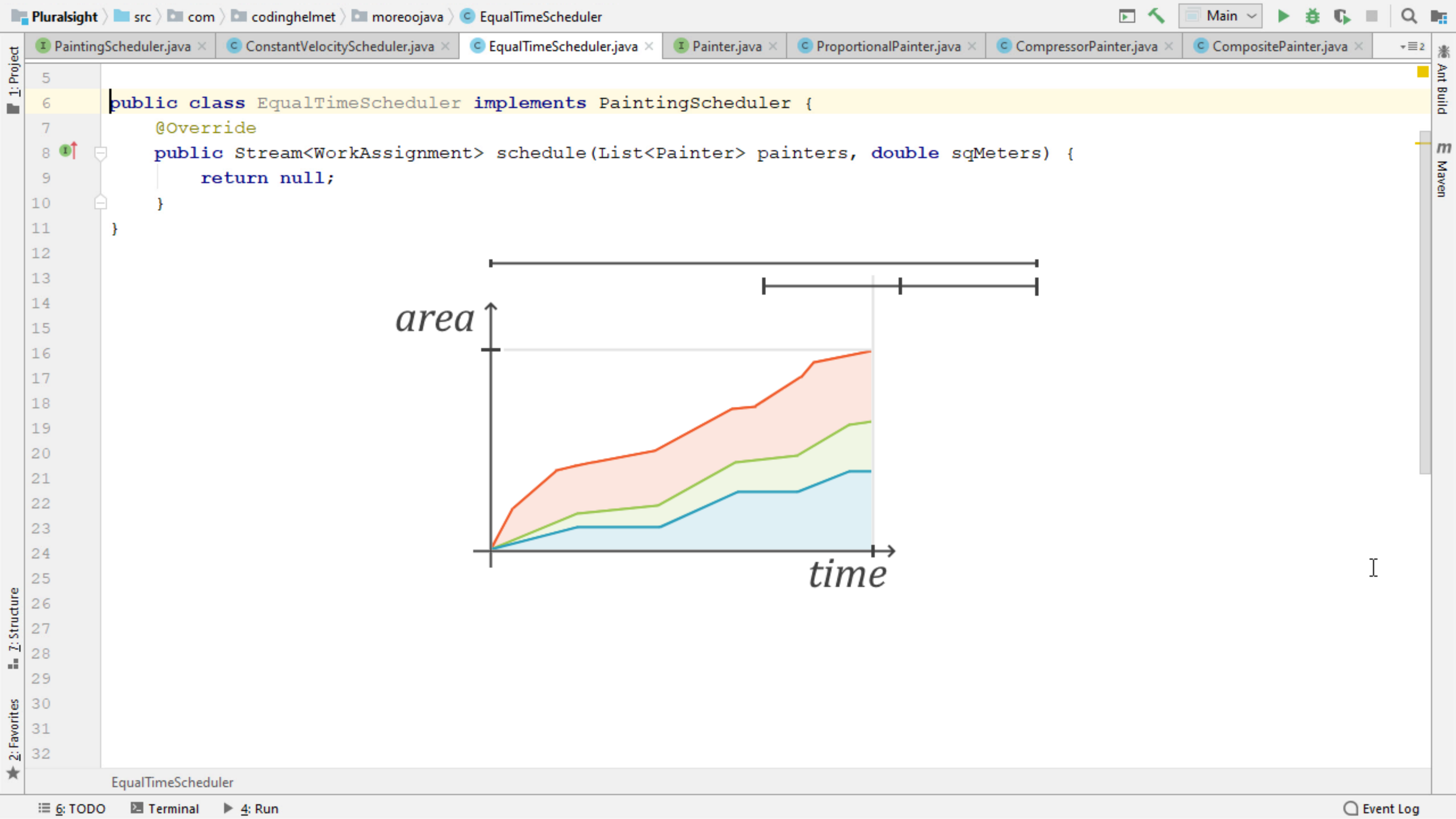
```
take available @painters
find fastest one
let it paint @area
or else report no work done
```

Used by customers



```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```

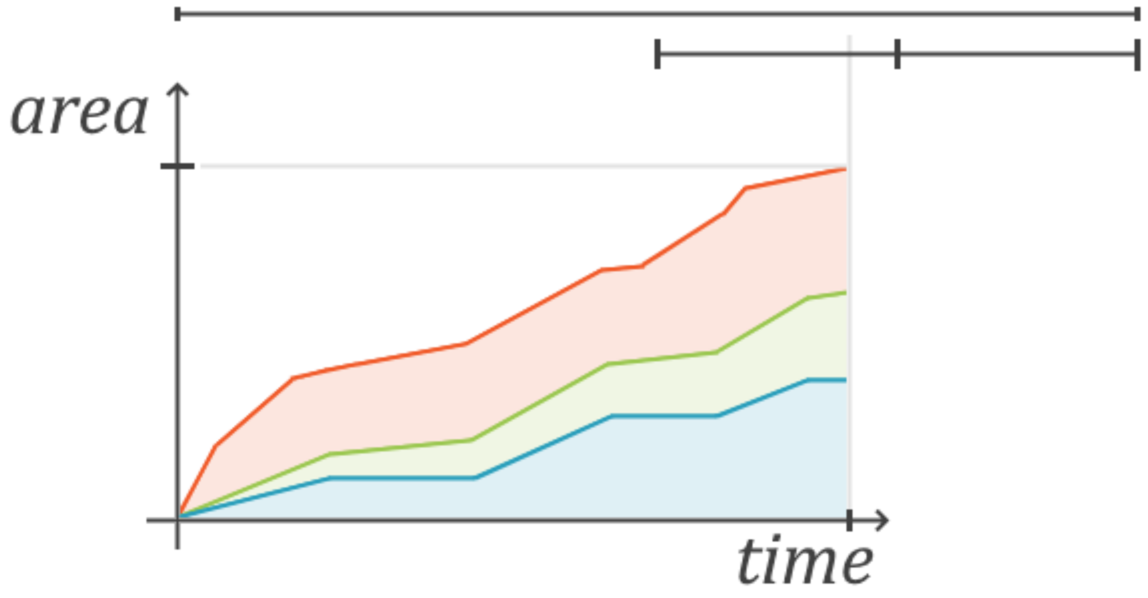




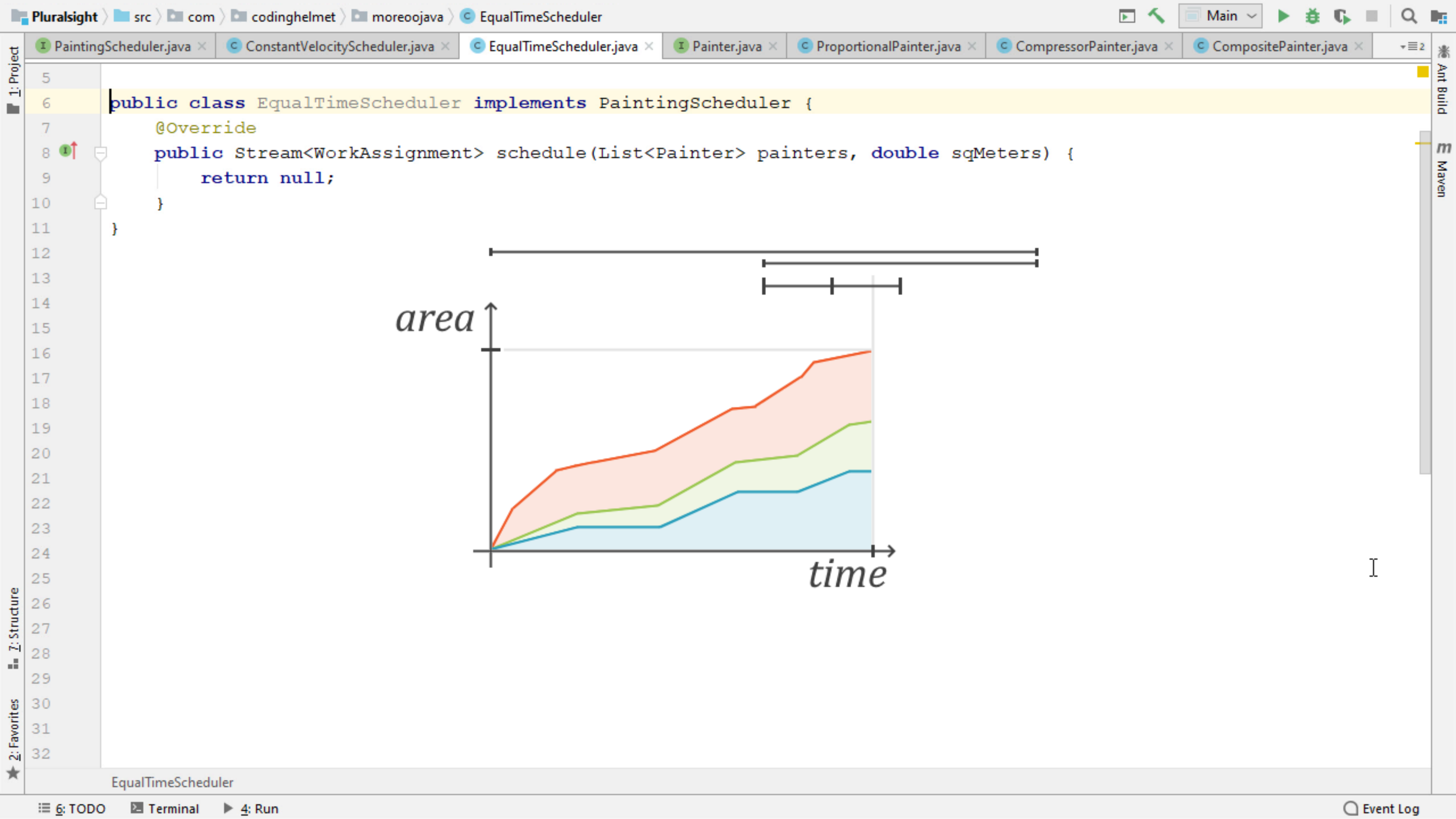
Pluralsight › src › com › codinghelmet › moreoojava › EqualTimeScheduler

PaintingScheduler.java | ConstantVelocityScheduler.java | EqualTimeScheduler.java | Painter.java | ProportionalPainter.java | CompressorPainter.java | CompositePainter.java

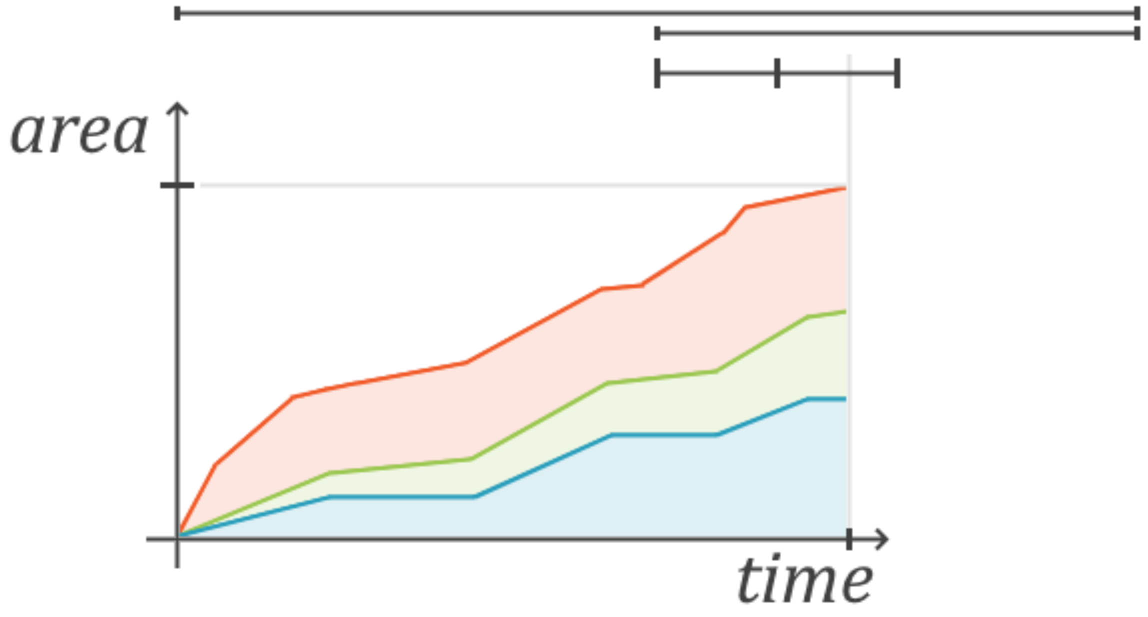
```java

public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```



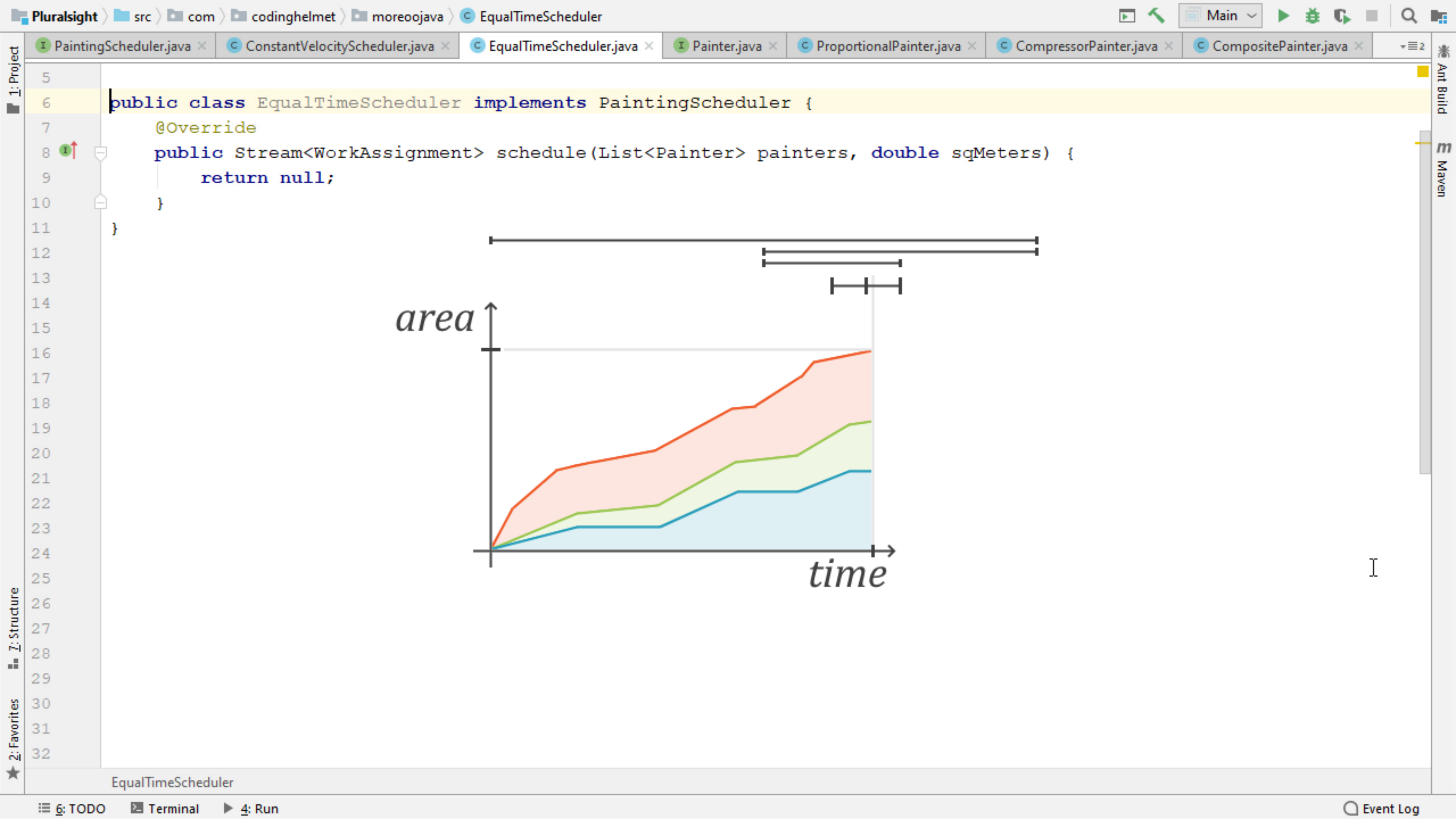
EqualTimeScheduler



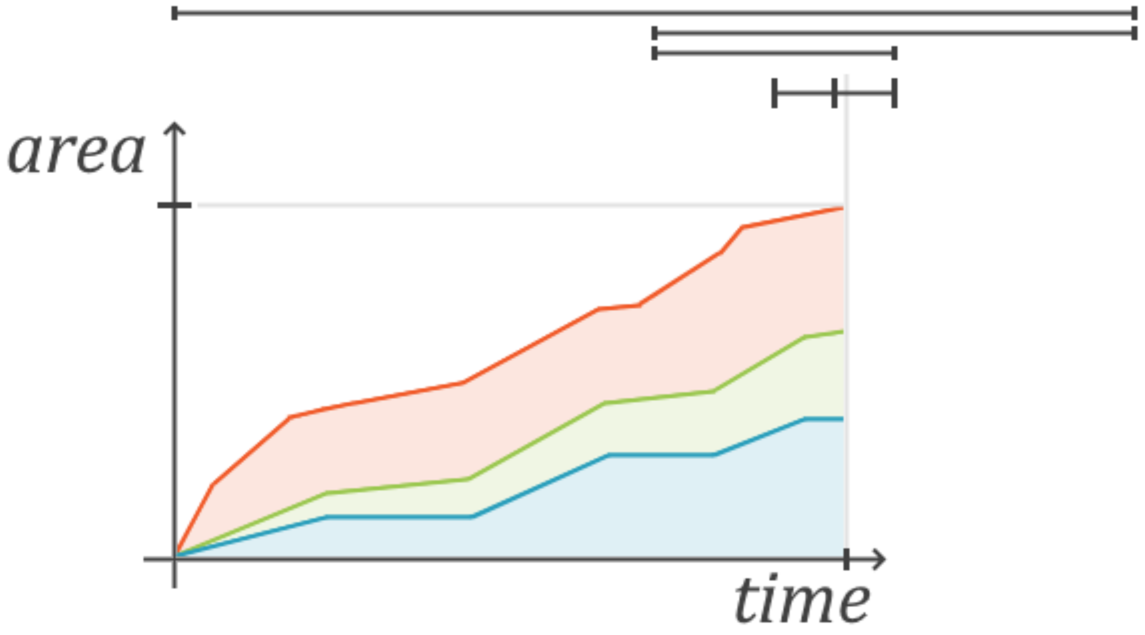
```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```

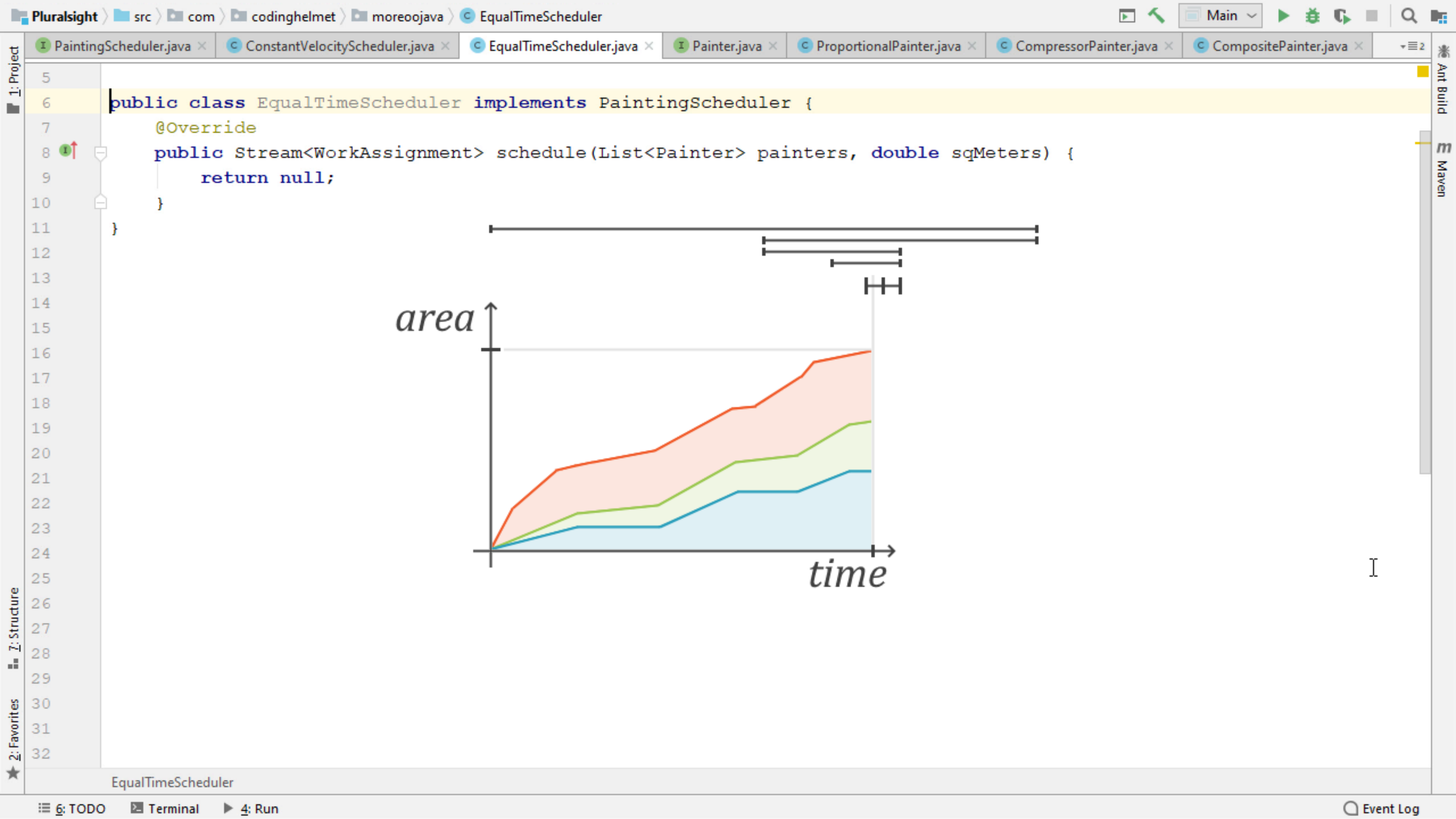


EqualTimeScheduler

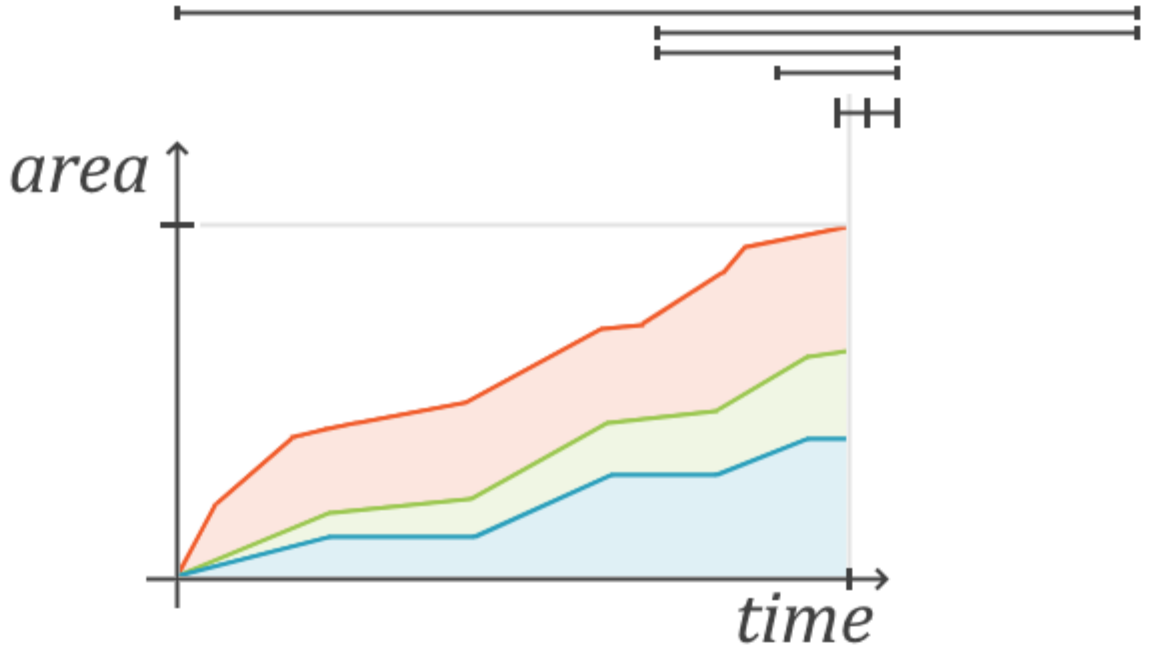


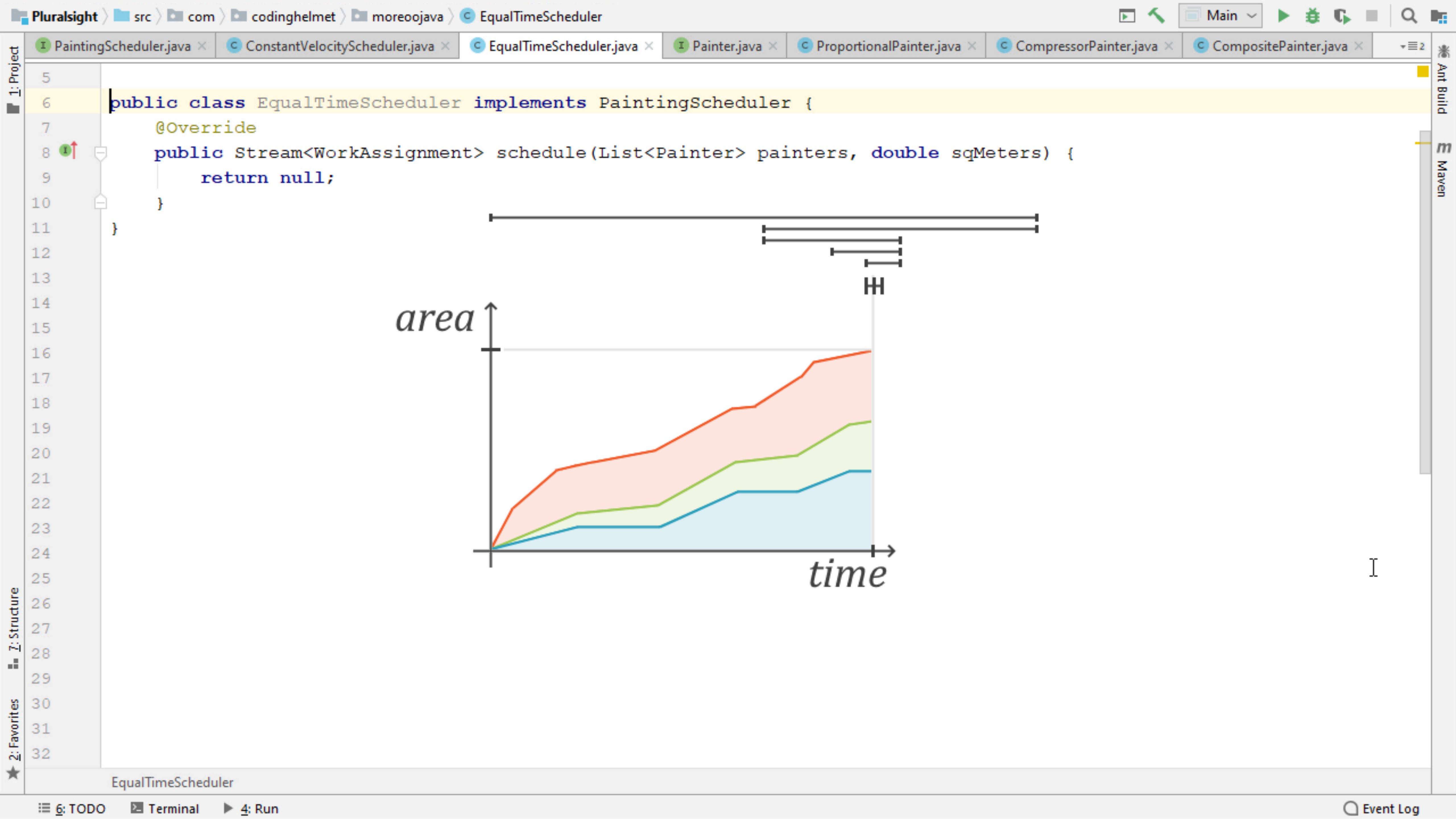
```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```



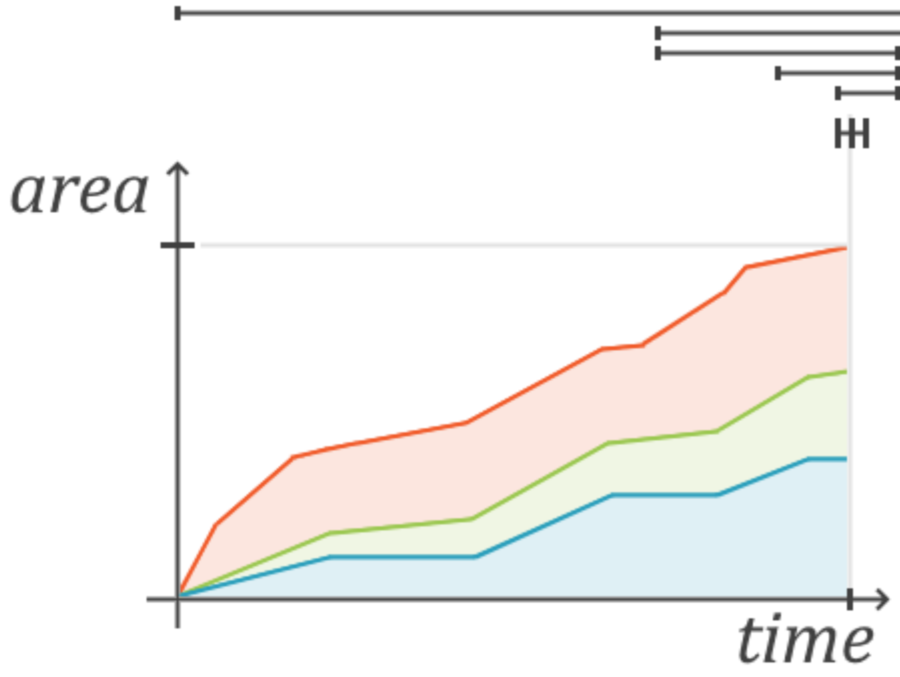


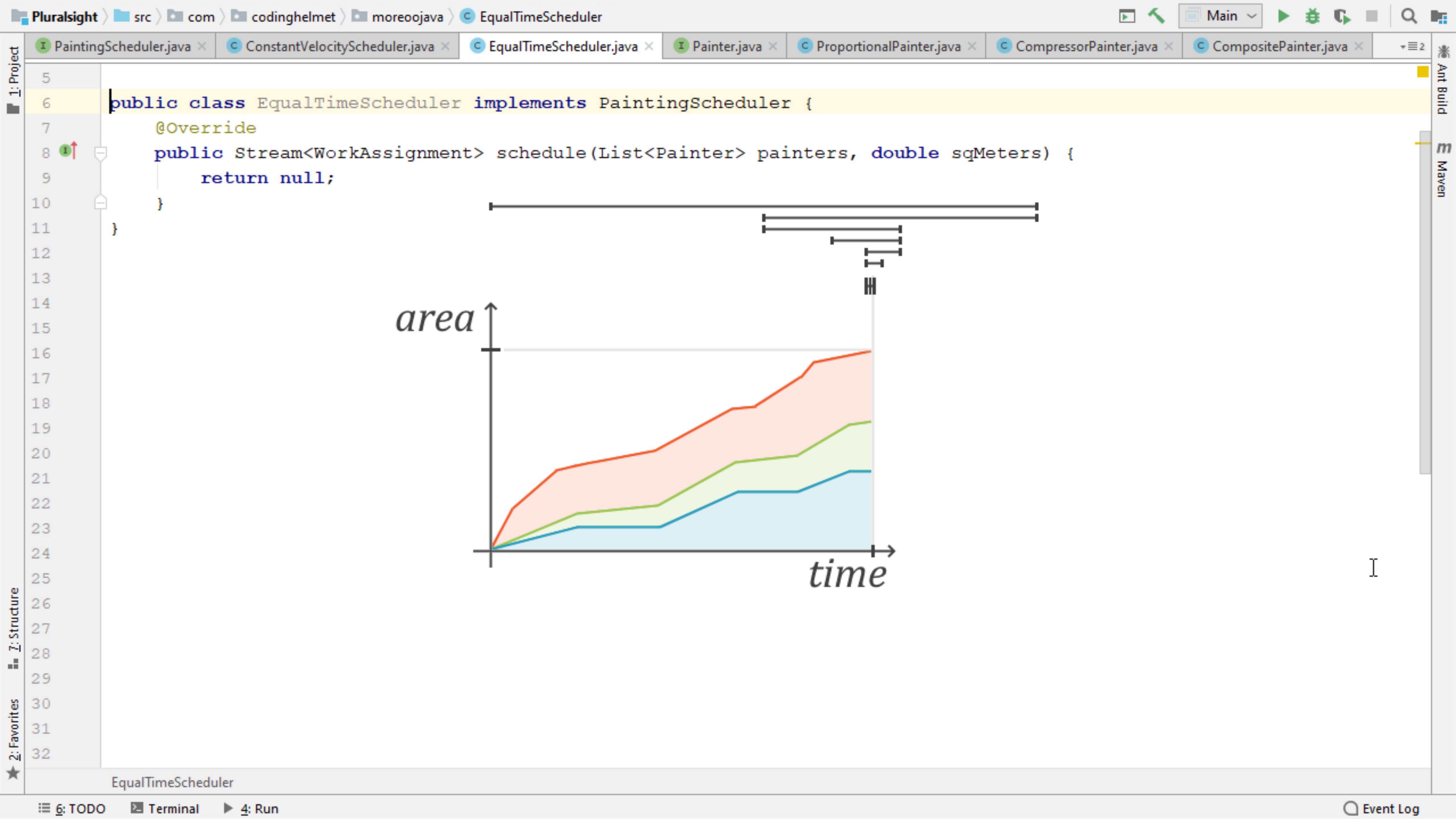
```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```



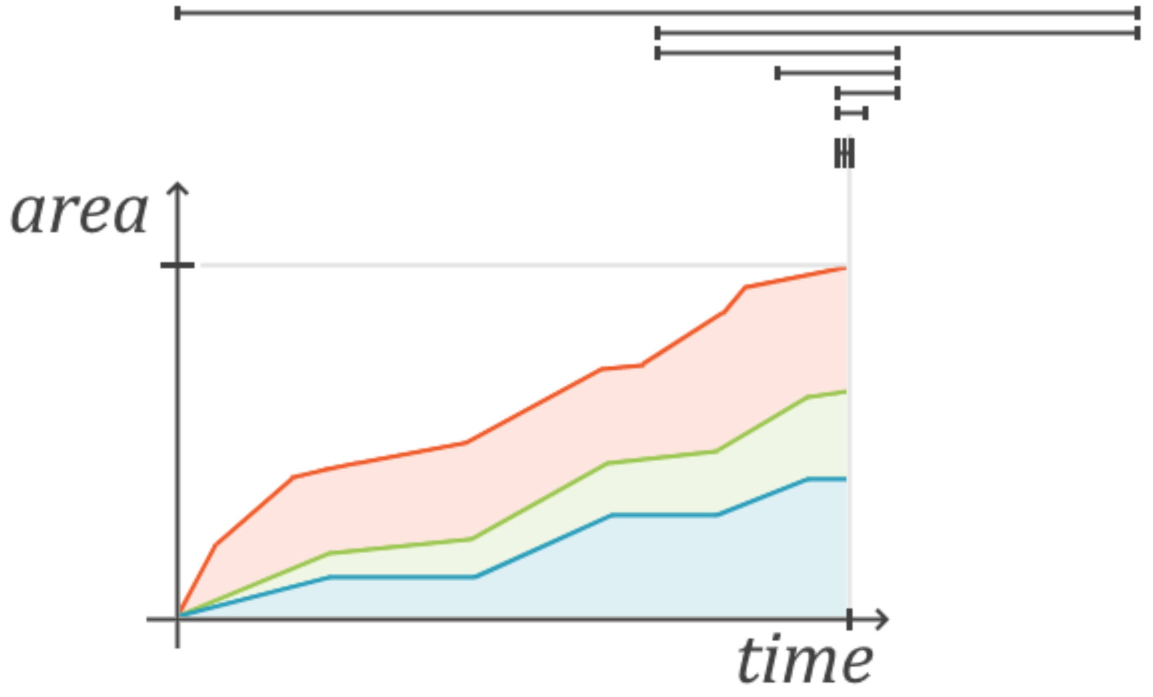


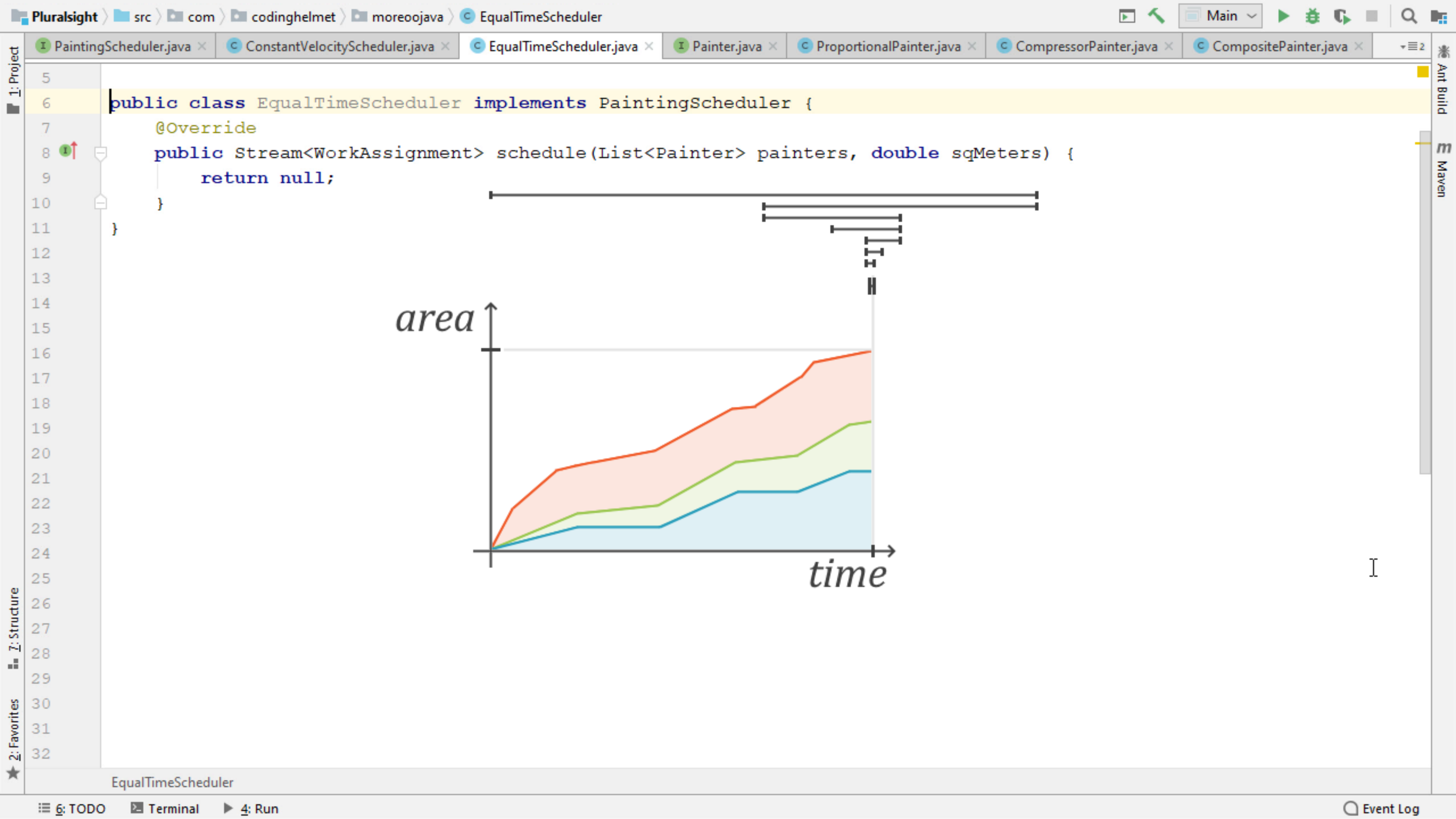
```java

public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```





```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```

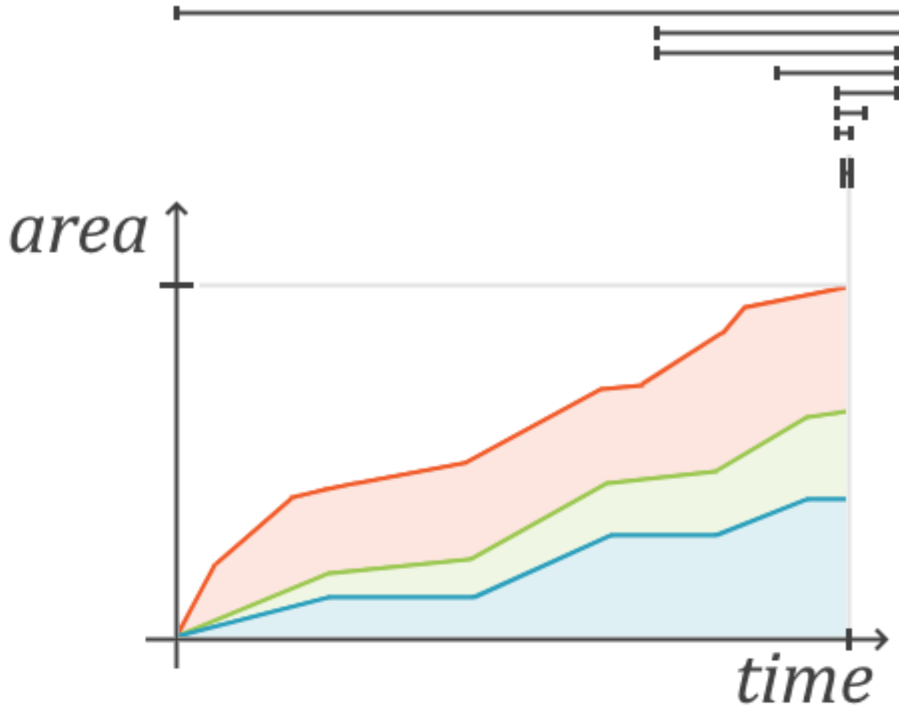




Pluralsight › src › com › codinghelmet › moreoojava › EqualTimeScheduler

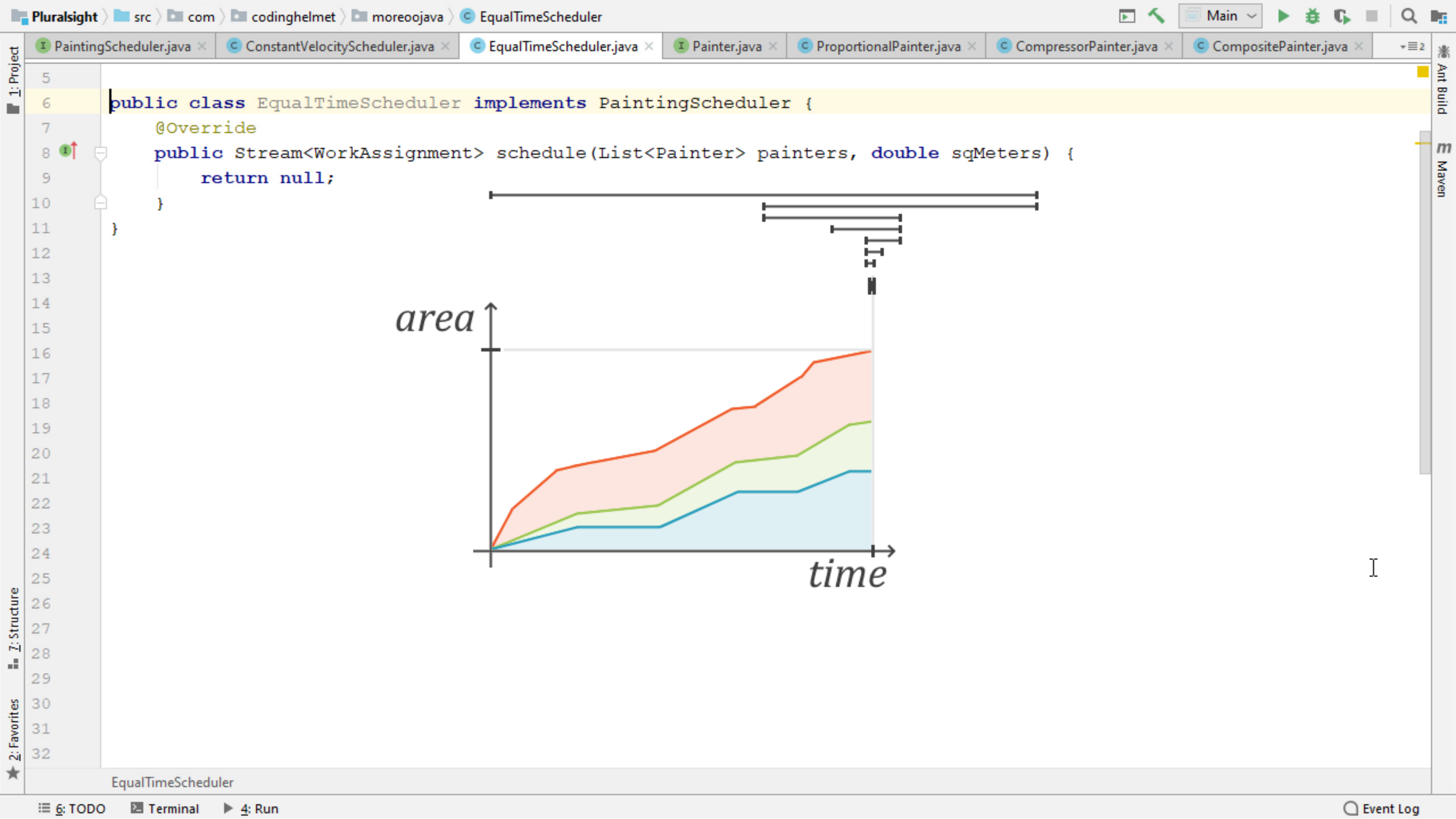
PaintingScheduler.java | ConstantVelocityScheduler.java | EqualTimeScheduler.java | Painter.java | ProportionalPainter.java | CompressorPainter.java | CompositePainter.java

```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```



EqualTimeScheduler

6: TODO | Terminal | 4: Run | Event Log

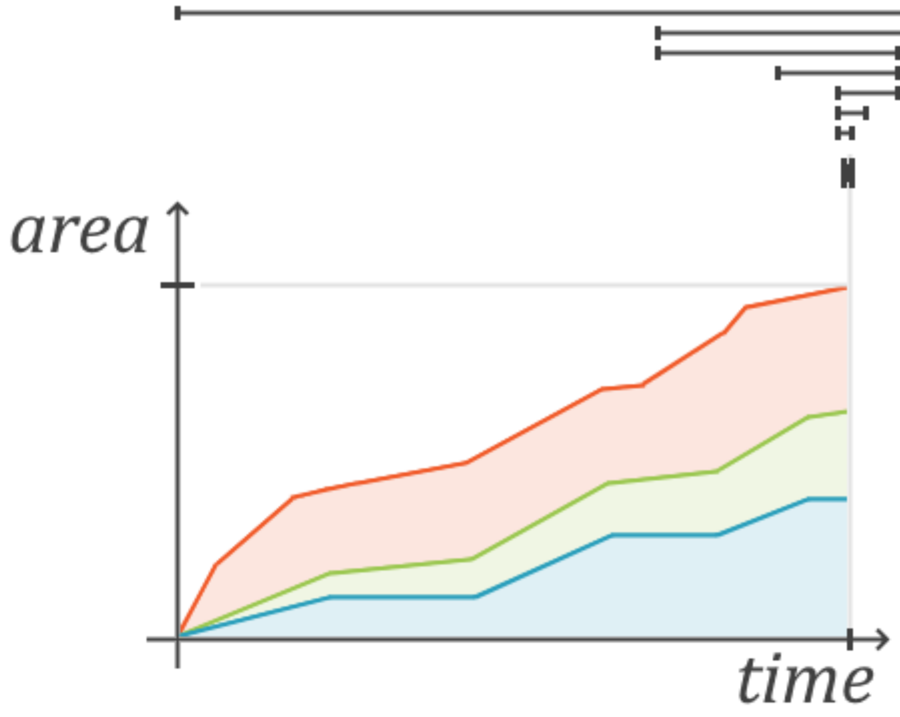


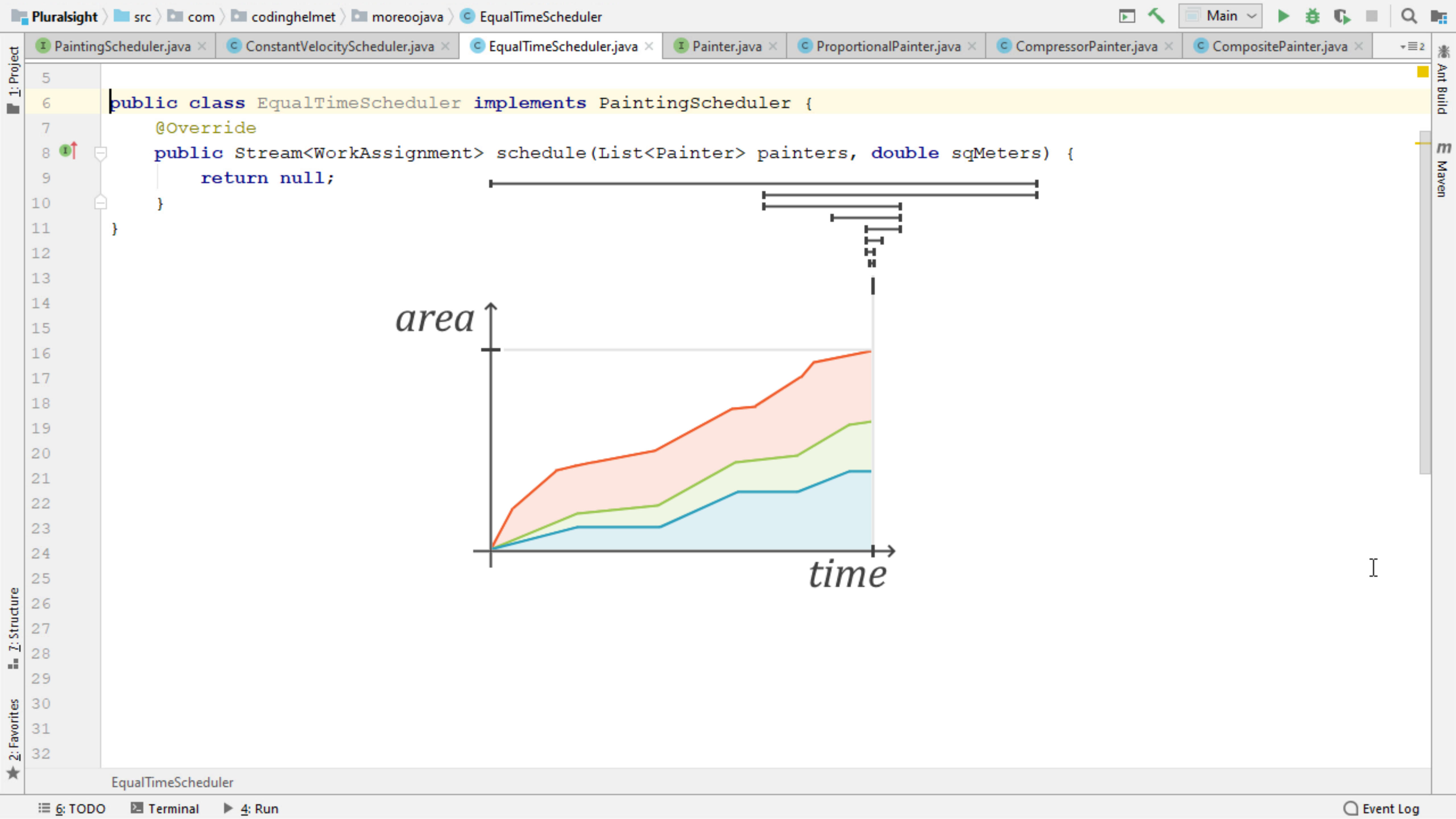
Pluralsight › src › com › codinghelmet › moreoojava › EqualTimeScheduler

PaintingScheduler.java | ConstantVelocityScheduler.java | EqualTimeScheduler.java | Painter.java | ProportionalPainter.java | CompressorPainter.java | CompositePainter.java

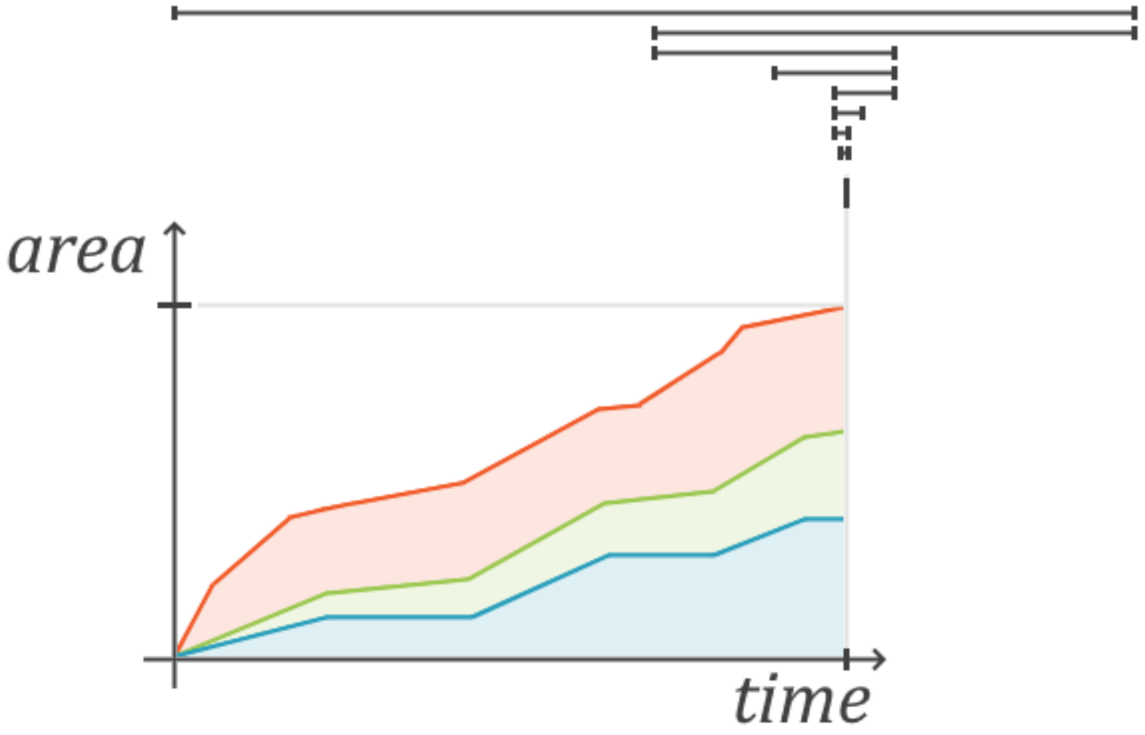
```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```

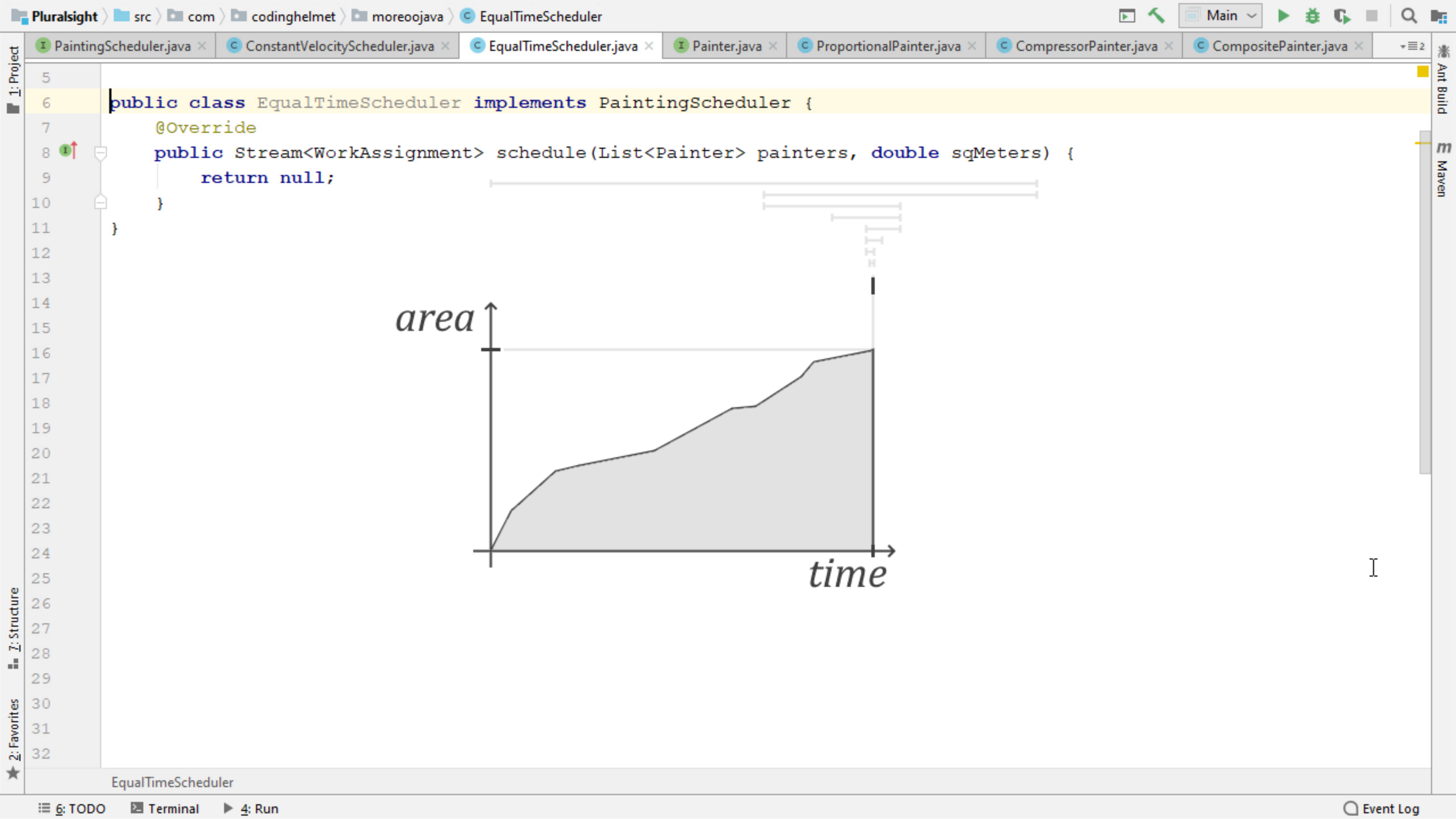
EqualTimeScheduler



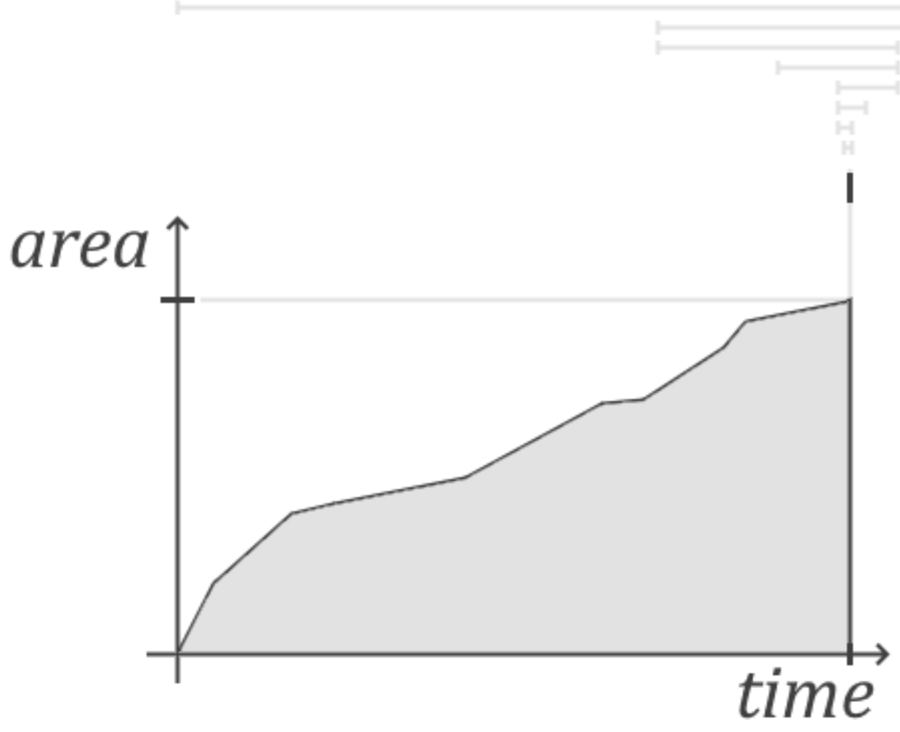


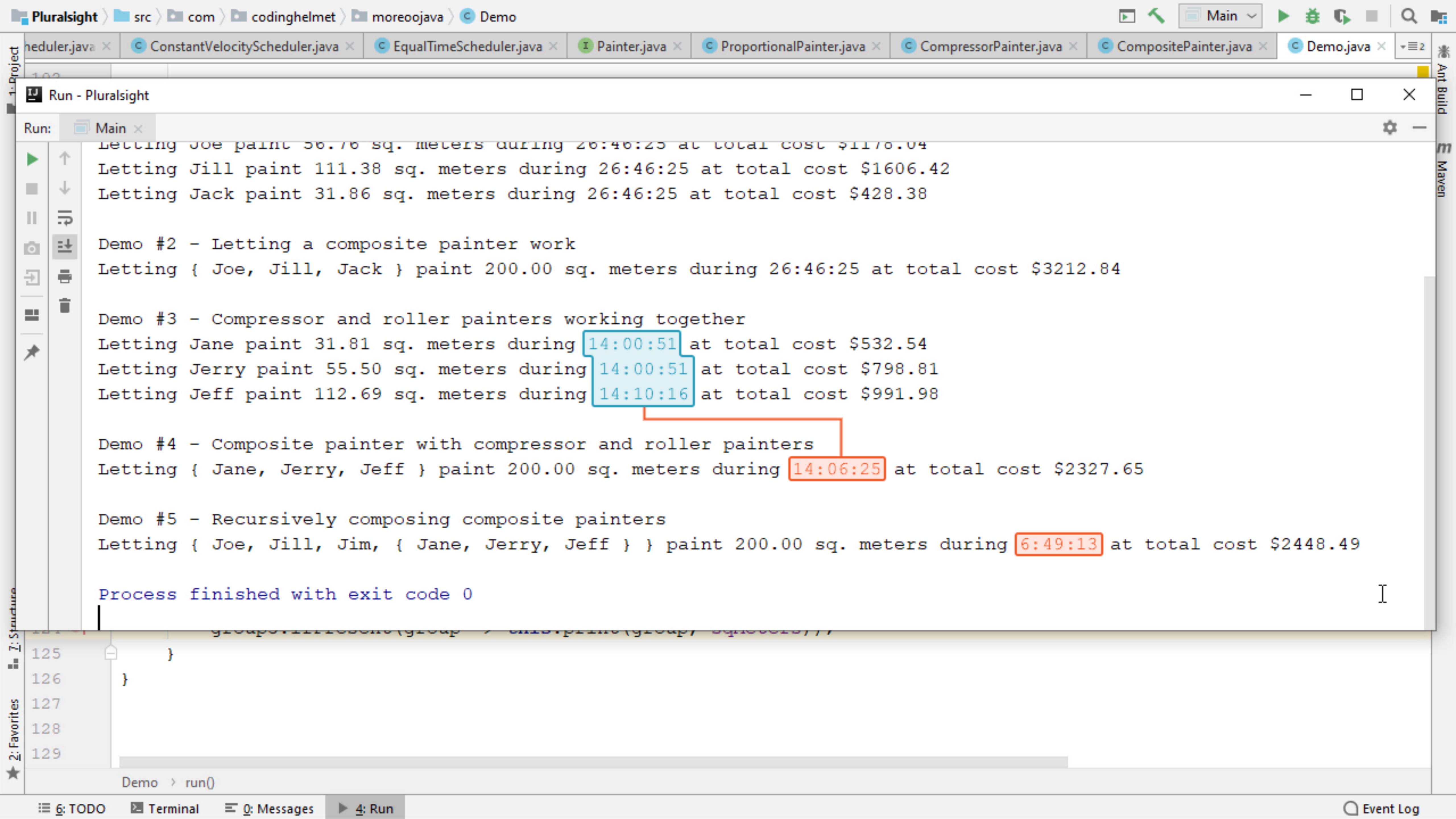
```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```





```java
public class EqualTimeScheduler implements PaintingScheduler {
    @Override
    public Stream<WorkAssignment> schedule(List<Painter> painters, double sqMeters) {
        return null;
    }
}
```





```
Letting Jill paint 111.38 sq. meters during 26:46:25 at total cost $1606.42
Letting Jack paint 31.86 sq. meters during 26:46:25 at total cost $428.38

Demo #2 - Letting a composite painter work
Letting { Joe, Jill, Jack } paint 200.00 sq. meters during 26:46:25 at total cost $3212.84

Demo #3 - Compressor and roller painters working together
Letting Jane paint 31.81 sq. meters during 14:00:51 at total cost $532.54
Letting Jerry paint 55.50 sq. meters during 14:00:51 at total cost $798.81
Letting Jeff paint 112.69 sq. meters during 14:10:16 at total cost $991.98

Demo #4 - Composite painter with compressor and roller painters
Letting { Jane, Jerry, Jeff } paint 200.00 sq. meters during 14:06:25 at total cost $2327.65

Demo #5 - Recursively composing composite painters
Letting { Joe, Jill, Jim, { Jane, Jerry, Jeff } } paint 200.00 sq. meters during 6:49:13 at total cost $2448.49

Process finished with exit code 0
```

```
    }
}
```

Summary

Expanding the deep domain model

- Inventing a domain-specific language
- Demonstrated an internal DSL
- Expressed in the programming language

Summary

Consuming an internal DSL

- Used by programmers on the project
- Lets them chain atomic transforms
- Leads to more expressive code
- Code is intention-revealing by design

Summary

Preconditions to developing a DSL

- Objects and methods must be composable
- Each operation is small and isolated
- Operations return composable objects
- Chaining atomic transforms to build complex behavior

Summary

Implementing composability

- Avoiding mutable methods
- Methods construct new objects
- Heavy use of immutable objects

Summary

Next module:

Increasing Flexibility by Removing Enums and Switch Statements