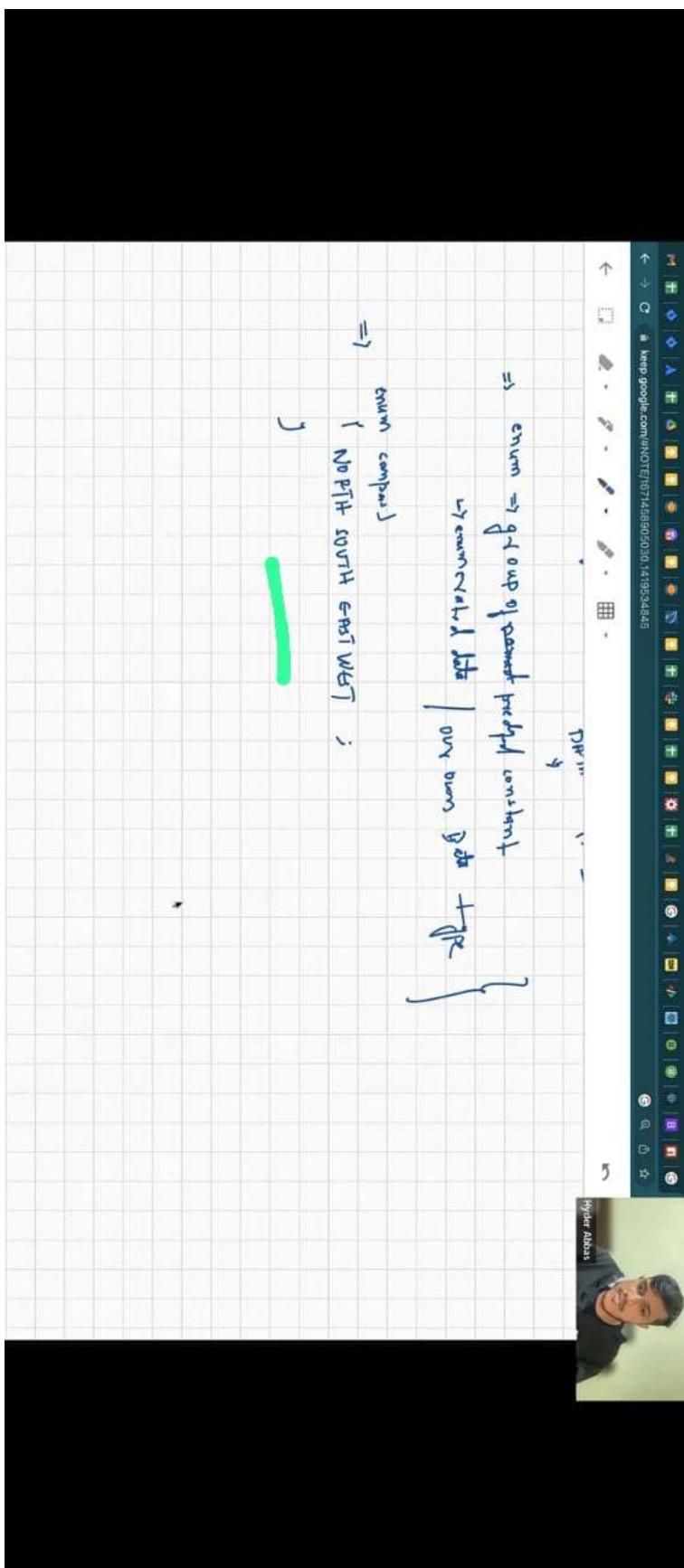


Java_Enum_And_Annotations



```
 4 //MALE FEMALE OTHERS
 5 // SUN MON TUE WED THU FRI SAT
 6 //JAN FEB MAR ----- DEC
 7 // PASS FAIL NR
 8 // so on.....
 9
10 enum Result
11 {
12     PASS, FAIL, NR; // static final
13 }
14 enum Gender
15 {
16     MALE, FEMALE, OTHERS;
17 }
18 enum Compass
19 {
20     NORTH, SOUTH, EAST, WEST;
21 }
22
23 public class LaunchEnum {
24
25     public static void main(String[] args) {
26         // TODO Auto-generated method stub
27
28     }
29 }
```





```
1 // Package Explorer X
2 // Java Project X
3 // Classpath
4 // Buildpath
5 // Output
6 // Errors
7 // Warnings
8 // Problems
9 // Tasks
10 // External Resources
11 // Launch Configurations
12 // Launching
13 // Run Configuration
14 // Run Configuration
15 // Run Configuration
16 // Run Configuration
17 // Run Configuration
18 // Run Configuration
19 // Run Configuration
20 // Run Configuration
21 // Run Configuration
22 // Run Configuration
23 // Run Configuration
24 // Run Configuration
25 // Run Configuration
26 // Run Configuration
27 // Run Configuration
```

Hy der Abbaus

```
1 //NORTH SOUTH EAST WEST
2 //MALE FEMALE OTHERS
3 // SUN MON TUE WED THU FRI SAT
4 //JAN FEB MAR ----- DEC
5 // PASS FAIL NR
6 // so on.....
7
8
9
10 enum Result
11 {
12     PASS, FAIL, NR; // static final
13     // fields --> instance var --> properties
14     // methods
15     //Constructor
16 }
17 //enum Gender
18 //{
19     MALE, FEMALE, OTHERS;
20 //}
21 //enum Compass
22 //{
23     NORTH, SOUTH, EAST, WEST;
24 //}
25
26 //class Demo
27 //
```



```
1 Package Explorer X 2 Playwing 3 Keygen 4 ChinesischeWelt 5 Darkjones 6 Dudo 7 Dudo; 8 Drunwell 9 Drunwell; 10 EuchseisErklaering 11 Erne 12 Inherita 13 Inherita; 14 JH System LaunchEnum2.java 15 JH System LaunchEnum2.java 16 JH System LaunchEnum2.java 17 JH System LaunchEnum2.java 18 { 19 public class LaunchEnum2 20 { 21     public static void main(String[] args) 22     { 23         Result res=Result.PASS; 24         System.out.println(res); 25     } 26 } 27 }
```

UZI AWI

Wrinkle Smartmirror 14:29:278

```
//PASS --> public static final Result PASS=new Result()
//FAIL --> static final Result FAIL=new Result()
//NR ----> static final Result NR=new Result();
Result()
{
    System.out.println("Constructor is called");
}
14 }
```

Console

Administrator: Lernumgebung für die Hochschule Anhalt

Constructor is called
Constructor is called
Constructor is called

Result()

System.out.println("Constructor is called");

14 }

5

6

7

8

9

10.

11

12

13

14 }

Java-IDE 17-Minute.11 Ks

```
 1  package Explorer;
 2
 3  import java.awt.*;
 4  import javax.swing.*;
 5  import java.awt.event.*;
 6  import java.util.*;
 7  import java.io.*;
 8  import java.net.*;
 9
10  public class LaunchEnum2 {
11      public static void main(String[] args) {
12          Result res=Result.PASS;
13          System.out.println(res);
14      }
15  }
16
17  public class LaunchEnum2 {
18      public static void main(String[] args) {
19          Result res=Result.FAIL;
20          System.out.println(res);
21          System.out.println("Constructor is called");
22          System.out.PASS;
23          System.out.println(res);
24          System.out.println("LaunchEnum2");
25          System.out.println("LaunchEnum2");
26          System.out.println("LaunchEnum2");
27          Result resArr[] = Result.FAIL.values();
28          for (Result res : resArr) {
29              System.out.println(res);
30          }
31      }
32  }
33
34
35  }
36
```

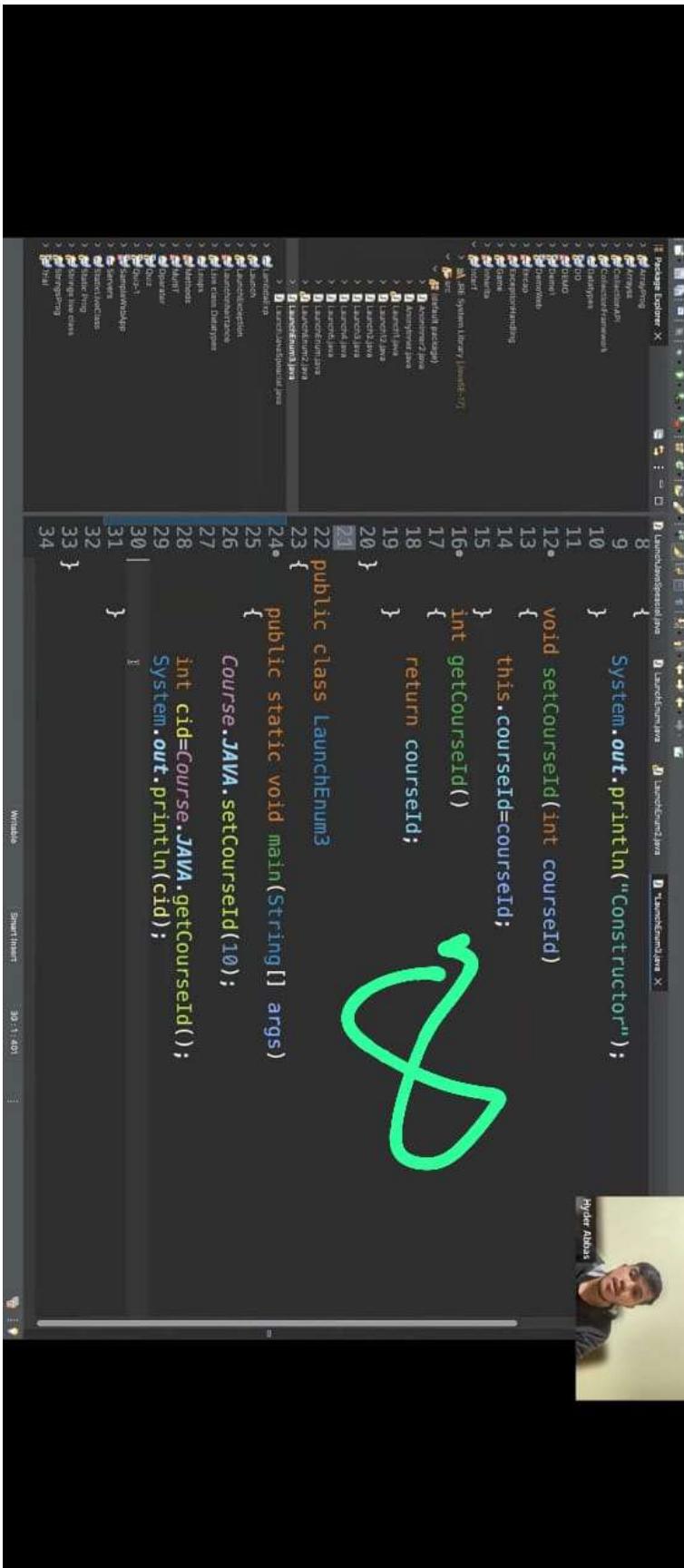


By der Abba

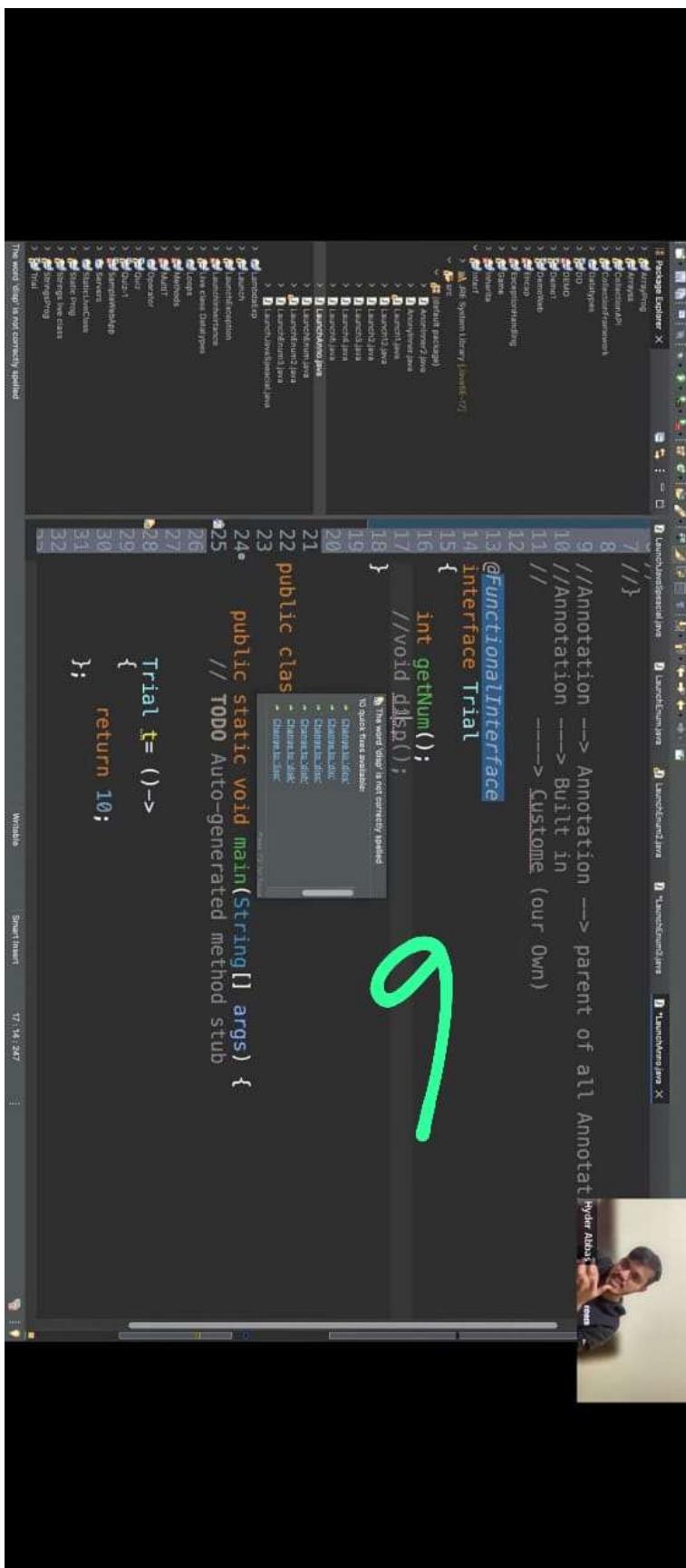
The screenshot shows a Java code editor with the following code:

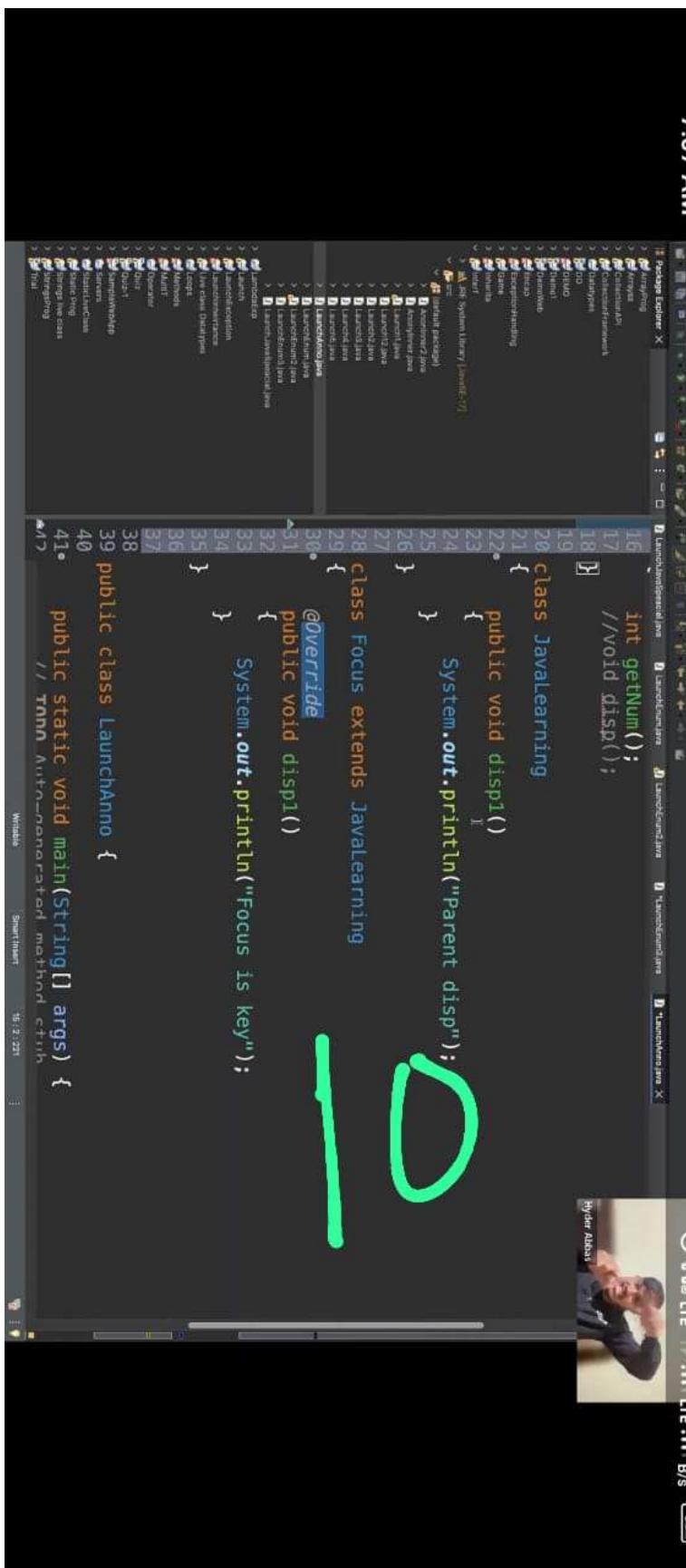
```
1 package Explorer;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class LaunchEnum3 {
7     public static void main(String[] args) {
8         public class Course {
9             int courseId;
10            Course() {
11                System.out.println("Constructor");
12            }
13            void setCourseId(int courseId) {
14                this.courseId=courseId;
15            }
16            int getCourseId() {
17                return courseId;
18            }
19        }
20    }
21 }
22 }
23 }
24 }
25 public class LaunchEnum3 {
26
27 }
```

A large green curly brace annotation is placed over the opening brace of the `Course` class definition at line 8. A small video thumbnail of a person is visible in the bottom right corner of the IDE window.



```
1  package Explorer;
2
3  import java.util.Scanner;
4
5  public class Explorer {
6      public static void main(String[] args) {
7          Scanner scanner = new Scanner(System.in);
8          System.out.println("Willkommen im Explorer!");
9          System.out.println("Was möchtest du machen?");
10         String command = scanner.nextLine();
11         if ("Dateien öffnen".equals(command)) {
12             System.out.println("Welche Datei möchtest du öffnen?");
13             String fileName = scanner.nextLine();
14             System.out.println("Du hast die Datei " + fileName + " geöffnet.");
15         } else if ("Dateien speichern".equals(command)) {
16             System.out.println("Welche Datei möchtest du speichern?");
17             String fileName = scanner.nextLine();
18             System.out.println("Du hast die Datei " + fileName + " gespeichert.");
19         } else if ("Dateien drucken".equals(command)) {
20             System.out.println("Welche Datei möchtest du drucken?");
21             String fileName = scanner.nextLine();
22             System.out.println("Du hast die Datei " + fileName + " gedruckt.");
23         } else if ("Dateien löschen".equals(command)) {
24             System.out.println("Welche Datei möchtest du löschen?");
25             String fileName = scanner.nextLine();
26             System.out.println("Du hast die Datei " + fileName + " gelöscht.");
27         } else if ("Dateien aktualisieren".equals(command)) {
28             System.out.println("Welche Datei möchtest du aktualisieren?");
29             String fileName = scanner.nextLine();
30             System.out.println("Du hast die Datei " + fileName + " aktualisiert.");
31         } else if ("Dateien aktualisieren".equals(command)) {
32             System.out.println("Welche Datei möchtest du aktualisieren?");
33             String fileName = scanner.nextLine();
34         }
35     }
36 }
```

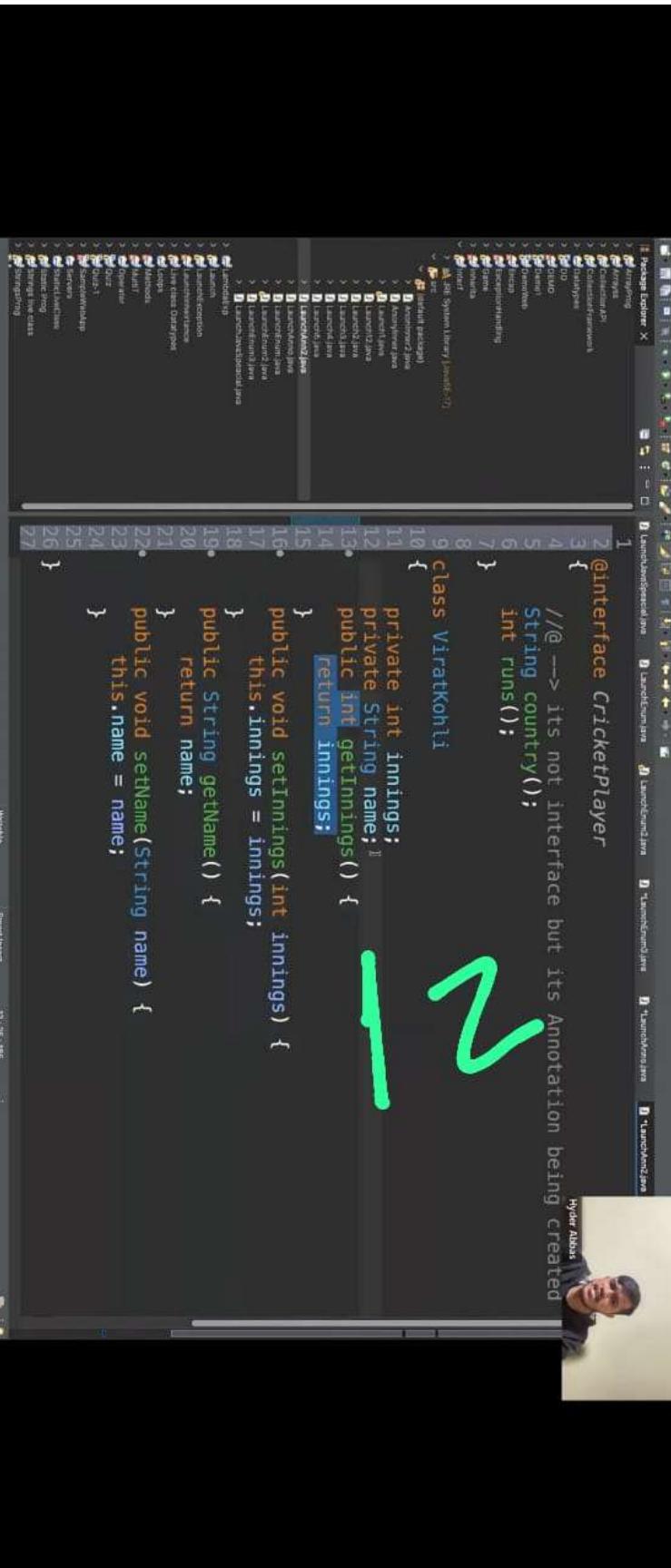




The screenshot shows a Java development environment with several windows open. The main window displays the following Java code:

```
1 // Annotation
2 // @CricketPlayer
3 //class Vlat
4 //{
5 //
6 //
7 //}
8
9 //Annotation --> Annotation --> parent of all Annotations
10 //Annotation ----> Built in
11 // -----> Custom (our own)
12
13
14 // Annotation -->
15 // (1) class
16 // (2) interface
17 // (3) method
18 // (4) fields // instance var
19 // (5) local variables
20 // (6) constructor
21 // (7) parameters
22 // (8) enum
23
24 @Function
25 interface
26 {
27     in
```

Annotations are highlighted with green underlines. A video player window titled "Hyper Abstraction" is visible in the bottom right corner.



```
1 package Explorer X
2 
3 import
4 import
5 import
6 import
7 import
8 import
9 import
10 import
11 import
12 import
13 import
14 import
15 import
16 import
17 import
18 import
19 import
20 import
21 import
22 import
23 import
24 import
25 import
26 import
27 import
```

```
1 @interface CricketPlayer
2 {
3     //(@--> its not interface but its Annotation being created
4     String country();
5     int runs();
6 }
7 }
```

```
8 class ViratKohli
9 {
10     private int innings;
11     private String name;
12     public int getInnings() {
13         return innings;
14     }
15     public void setInnings(int innings) {
16         this.innings = innings;
17     }
18     public String getName() {
19         return name;
20     }
21     public void setName(String name) {
22         this.name = name;
23     }
24 }
```



```
1 package Explorer X;
2
3 import java.awt.*;
4 import javax.swing.*;
5 import javax.swing.border.*;
6 import javax.swing.table.*;
7 import javax.swing.tree.*;
8 import javax.swing.GroupLayout.*;
9 import javax.swing.LayoutStyle.*;
10 import javax.swing.JButton.*;
11 import javax.swing.JList.*;
12 import javax.swing.JPanel.*;
13 import javax.swing.JScrollPane.*;
14 import javax.swing.JTabbedPane.*;
15 import javax.swing.JTable.*;
16 import javax.swing.JTextPane.*;
17 import javax.swing.JTree.*;
18 import javax.swing.JTextArea.*;
19 import javax.swing.JComponent.*;
20 import javax.swing.JEditorPane.*;
21 import javax.swing.JTextPane.*;
22 import javax.swing.JTextPane.*;
23 import javax.swing.JTextPane.*;
24 import javax.swing.JTextPane.*;
25 import javax.swing.JTextPane.*;
26 import javax.swing.JTextPane.*;
27 import javax.swing.JTextPane.*;
```

The code block shows a Java application window with a file tree on the left and a code editor on the right. The code is annotated with numbers 1 through 27. A large green hand-drawn mark is drawn over the code area, covering lines 12 through 17.

The screenshot shows an IDE interface with a code editor containing Java code. A green checkmark is drawn over the code at line 15.

```
1  @TargetElementType T)
2  @interface CricketPlayer {
3  {
4      // ---> its not
5      // String country()
6      // int runs() defau
7
8      String country();
9      int runs();
10 }
11 //Target
12 //retention policy
13
14
15 @CricketPlayer(country="India", runs=2000)
16 class ViratKohli {
17 {
18     private int innings;
19     private String name;
20
21
22     public int getInnings() {
23         return innings;
24     }
25     public void setInnings(int innings) {
26         this.innings = innings;
27     }
}
```



```
1. Package Explorer X
  2. Lombok-Uberspecification
  3. LombokExample
  4. LombokExampleTest
  5. LombokExampleMain
  6. LombokExampleRun
  7. LombokExampleRun
  8. LombokExampleRun
  9. LombokExampleRun
  10. LombokExampleRun
  11. LombokExampleRun
  12. LombokExampleRun
  13. LombokExampleRun
  14. LombokExampleRun
  15. LombokExampleRun
  16. LombokExampleRun
  17. LombokExampleRun
  18. @CricketPlayer(country="India", runs=2000)
  19. class ViratKohli
  20. {
  21.     private int innings;
  22.     private String name;
  23. }
  24.
  25. public int getInnings() {
  26.     return innings;
  27. }
```

import java.lang.annotation.ElementType;
import java.lang.annotation.Target;

@Target(ElementType.TYPE)

//@ --> its not interface but its Annotation being created

String country();
int runs();

}/Target
//retention policy

@CricketPlayer(country="India", runs=2000)

class ViratKohli

private int innings;
private String name;

public int getInnings() {
 return innings;





```
1 Package Explorer X
2 
32 public void setInnings(int innings) {
33     this.innings = innings;
34 }
35 public String getName() {
36     return name;
37 }
38 public void setName(String name) {
39     this.name = name;
40 }
41 }
42 }
43 }
44 public class LaunchAnn2 {
45     public static void main(String[] args) {
46         VirratKohli vk=new VirratKohli();
47         vk.setInnings(300);
48         vk.setName("VK");
49     }
50 }
51 System.out.println(vk.getInnings());
52 System.out.println(vk.getName());
53 }
54 }
55 }
56 }
57 }
58 }
```

Wrinkle

Smart Insert

16 : 172 / 400



The image shows a person standing behind a desk, looking towards the camera. On the desk is a computer monitor displaying a Java code editor. The code is written in Java and includes annotations like `@Retention(RetentionPolicy.RUNTIME)`. A large green arrow points from the bottom left towards the code on the screen.

```
1 package Engine;
2
3 import java.awt.*;
4 import javax.swing.*;
5 import java.awt.event.*;
6 import java.awt.image.*;
7 import java.awt.FontMetrics.*;
8 import java.awt.Graphics.*;
9 import java.awt.Color.*;
10 import java.awt.AlphaComposite.*;
11 import java.awt.RenderingHints.*;
12 import java.awt.image.BufferedImage.*;
13 import java.awt.image.PixelGrabber.*;
14 import java.awt.image.WritableRaster.*;
15 import java.awt.image.Raster.*;
16 import java.awt.image.DataBuffer.*;
17 import java.awt.image.DataBufferByte.*;
18 import java.awt.image.DataBufferInt.*;
19 import java.awt.image.DataBufferFloat.*;
20 import java.awt.image.DataBufferBoolean.*;
21 import java.awt.image.DataBufferDouble.*;
22 import java.awt.image.DataBufferShort.*;
23 import java.awt.image.DataBufferLong.*;
24 import java.awt.image.DataBufferBoolean.*;
25 import java.awt.image.DataBufferBoolean.*;
26 import java.awt.image.DataBufferBoolean.*;
27 import java.awt.image.DataBufferBoolean.*;
28 import java.awt.image.DataBufferBoolean.*;
29 import java.awt.image.DataBufferBoolean.*;
30 import java.awt.image.DataBufferBoolean.*;
31 import java.awt.image.DataBufferBoolean.*;
32 import java.awt.image.DataBufferBoolean.*;
33 import java.awt.image.DataBufferBoolean.*;
34 import java.awt.image.DataBufferBoolean.*;
35 import java.awt.image.DataBufferBoolean.*;
36 import java.awt.image.DataBufferBoolean.*;
37 import java.awt.image.DataBufferBoolean.*;
38 import java.awt.image.DataBufferBoolean.*;
39 import java.awt.image.DataBufferBoolean.*;
40 import java.awt.image.DataBufferBoolean.*;
41 import java.awt.image.DataBufferBoolean.*;
42 import java.awt.image.DataBufferBoolean.*;
43 import java.awt.image.DataBufferBoolean.*;
44 import java.awt.image.DataBufferBoolean.*;
45 import java.awt.image.DataBufferBoolean.*;
46 import java.awt.image.DataBufferBoolean.*;
47 import java.awt.image.DataBufferBoolean.*;
48 import java.awt.image.DataBufferBoolean.*;
49 import java.awt.image.DataBufferBoolean.*;
50 import java.awt.image.DataBufferBoolean.*;
51 import java.awt.image.DataBufferBoolean.*;
52 import java.awt.image.DataBufferBoolean.*;
53 import java.awt.image.DataBufferBoolean.*;
54 import java.awt.image.DataBufferBoolean.*;
55 import java.awt.image.DataBufferBoolean.*;
56 import java.awt.image.DataBufferBoolean.*;
57 import java.awt.image.DataBufferBoolean.*;
58 import java.awt.image.DataBufferBoolean.*;
59 import java.awt.image.DataBufferBoolean.*;
60 import java.awt.image.DataBufferBoolean.*;
61 import java.awt.image.DataBufferBoolean.*;
62 import java.awt.image.DataBufferBoolean.*;
63 import java.awt.image.DataBufferBoolean.*;
64 import java.awt.image.DataBufferBoolean.*;
65 import java.awt.image.DataBufferBoolean.*;
66 import java.awt.image.DataBufferBoolean.*;
67 }
```



```
1 Package Explorer X
2 Java -> src-> com.abbas.abc-> CricketPlayer.java
3 Javadoc
4 Javadoc
5 Javadoc
6 Javadoc
7 Javadoc
8 Javadoc
9 Javadoc
10 Javadoc
11 Javadoc
12 Javadoc
13 Javadoc
14 Javadoc
15 Javadoc
16 Javadoc
17 Javadoc
18 Javadoc
19 Javadoc
20 Javadoc
21 Javadoc
22 Javadoc
23 Javadoc
24 Javadoc
25 Javadoc
26 Javadoc
27 Javadoc
```

18

```
1 import java.lang.annotation.Annotation;
2 import java.lang.annotation.ElementType;
3 import java.lang.annotation.Retention;
4 import java.lang.annotation.RetentionPolicy;
5 import java.lang.annotation.Target;
6
7 @Target(ElementType.TYPE, ElementType.METHOD, ElementType.LOCAL)
8 @Retention(RetentionPolicy.RUNTIME)
9
10 {
11     // @ --> its not interface but its Annotation being created
12     String country() default "India";
13     int runs() default 20000;
14
15     // String country();
16     // int runs();
17 }
18
19 //Target
20
21
22 // @CricketPlayer(country="India", runs=2000)
23
24 @CricketPlayer
25 class ViratKohli
26 {
27     private int innings;
```



```
1  Desktop Explorer X
2  20 // @Test
3  21 public void putL10n() {
4  22     // @CricketPlayer(country="India", runs=2000)
5  23
6  24     @CricketPlayer
7  25     class ViratKohli
8  26     {
9  27         private int innings;
10  28         private String name;
11  29
12  30     @CricketPlayer
13  31     public int getInnings() {
14  32         return innings;
15  33     }
16  34     public void setInnings(int innings) {
17  35         this.innings = innings;
18  36     }
19  37     public String getName() {
20         return name;
21     }
22     this.name = name;
23
24     public void setName(String name) {
25
26         this.name = name;
27     }
28
29     public class LaunchAnn2 {
30
31         public void launchAnn2() {
32             System.out.println("Launch Ann2");
33         }
34     }
35
36     public void launchAnn1() {
37         System.out.println("Launch Ann1");
38     }
39
40     public void launchAnn3() {
41         System.out.println("Launch Ann3");
42     }
43
44     public void launchAnn4() {
45     }
46 }
```