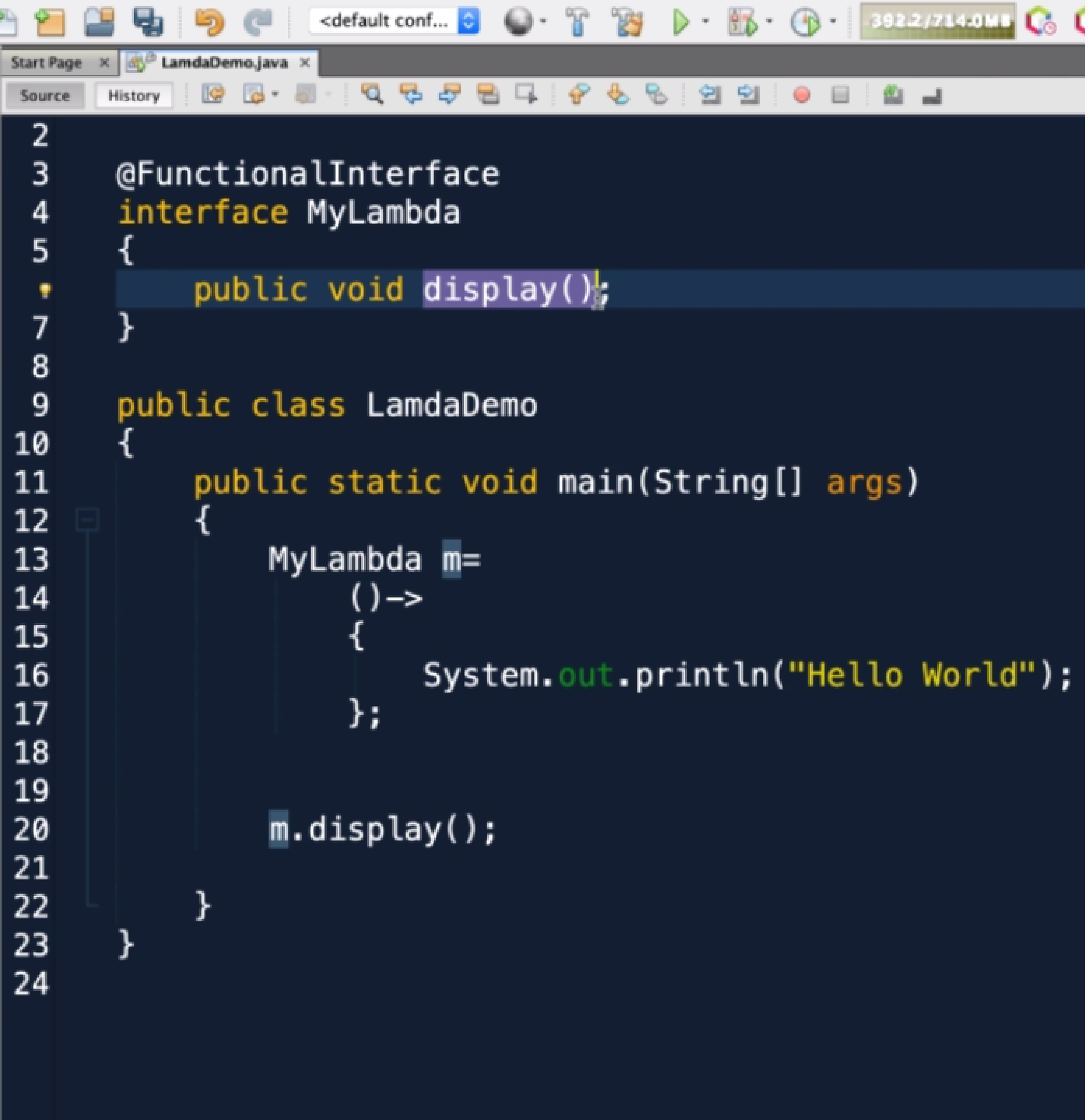


```java
@FunctionalInterface
interface MyLambda
{
    public void display();
}

public class LamdaDemo
{
    public static void main(String[] args)
    {
        MyLambda m=
            ()->
            {
                System.out.println("Hello World");
            };


        m.display();

    }
}
```

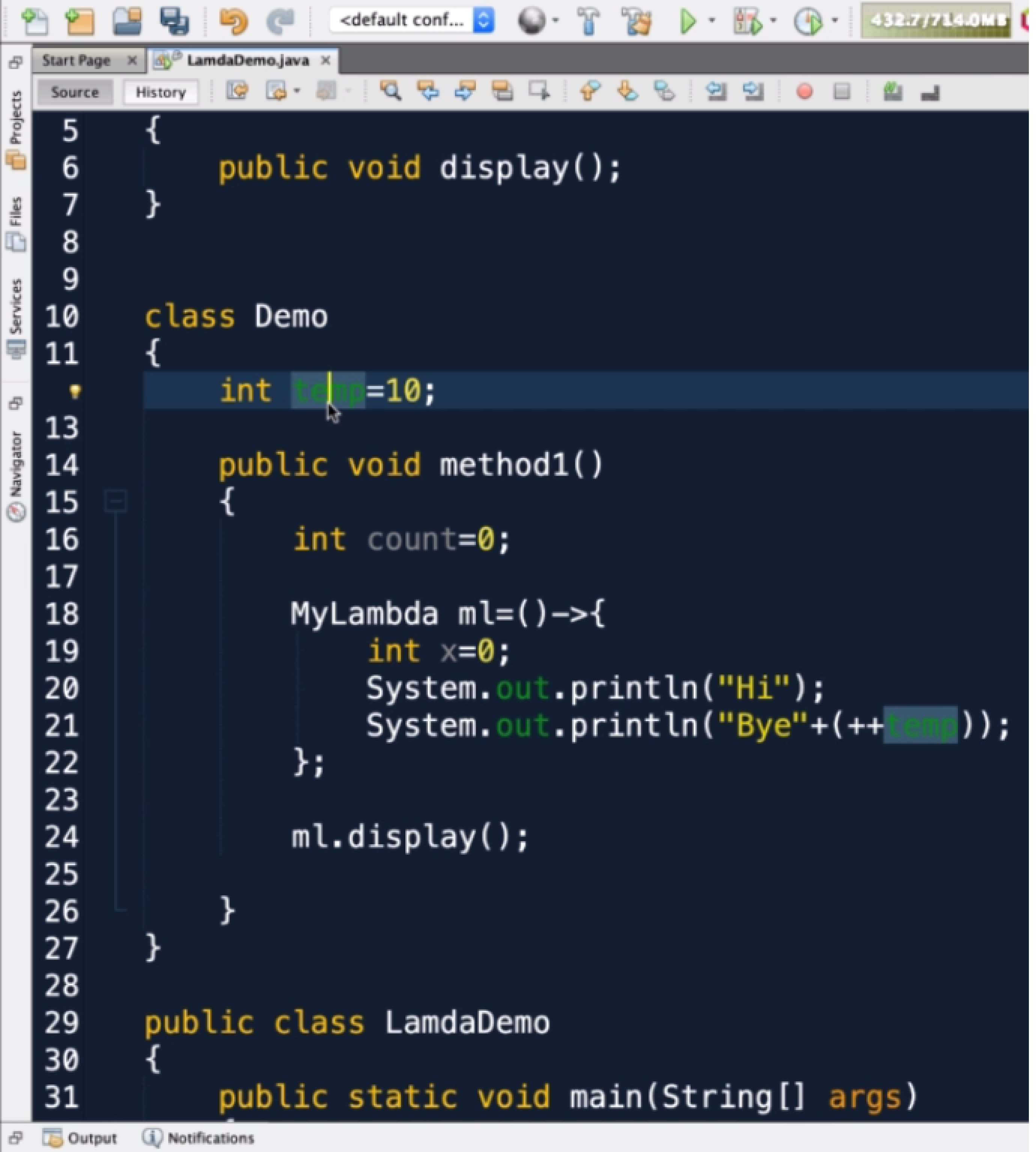
--> Java mein, Lambda Expression ek concise tareeka hai jo anonymous functions (jo ki bina naam ke functions hote hain) ko represent karne mai help karta hai .

--> Java 8 mein introduce hui, Lambda Expressions functional interfaces ke saath kaam karne ke liye syntax ko simplify karte hain.

--> Lambda expression ka anadar sirf ek hi method allowed hai.

--> Lambda expressions aksar functional interfaces ke saath istemal hote hain jismai sirf ek hi single abstract method hoti hain.

--> Lambda Expression ka basic syntax hota hai (parameters) -> { statements; }.



```
    {
        public void display();
    }


class Demo
{
        int temp=10;

        public void method1()
        {
            int count=0;

            MyLambda ml=()->{
                int x=0;
                System.out.println("Hi");
                System.out.println("Bye"+(++temp));
            };

            ml.display();

        }
}

public class LamdaDemo
{
        public static void main(String[] args)
```

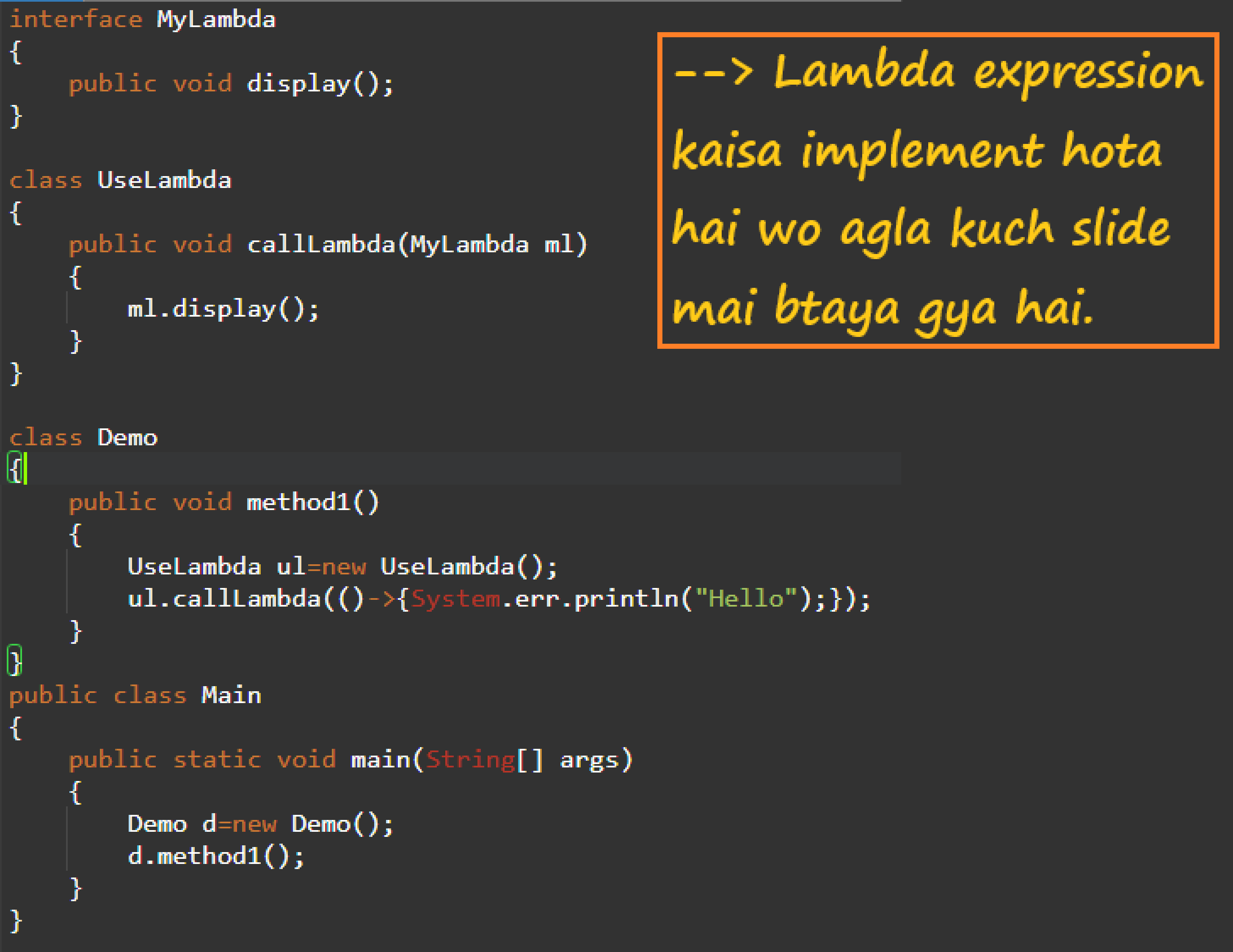
--> Ham lamda expression ka andar koi bhi variabale intialize kar sakta hai and uska use bhi kar sakta hai. Ex: int x;

--> ham lamda expressiion kai bhar aur method ka andar bhi variable ko intialize kar sakta hai and uska use bhi kar sakta hai inside expression.
Ex: int count=0;

Upper dia gya dona method mai, ham variable ka kuch bhi kar sakta hai bas update nahi kar sakta hai, nahi too fir error aayaga.

Also, agar hmna expression mai variable ko use kar liya hai and then within same method aagar ham usa update karta hai too bhi error hoga. Aisa iseliya hota hai kyuki expression can access only those variables which are final or effectively final.

--> Inside same class, agar ham koi global variable declare karta hai, too usa lamda expression ka andar update kiya ja sakta hai.
Ex: int temp =0;



```java
interface MyLambda
{
    public void display();
}

class UseLambda
{
    public void callLambda(MyLambda ml)
    {
        ml.display();
    }
}

class Demo
{
    public void method1()
    {
        UseLambda ul=new UseLambda();
        ul.callLambda(()->{System.err.println("Hello");});
    }
}
public class Main
{
    public static void main(String[] args)
    {
        Demo d=new Demo();
        d.method1();
    }
}
```

--> Lambda expression kaisa implement hota hai wo agla kuch slide mai btaya gya hai.

1. **Interface MyLambda:**

java Copy code

```java
interface MyLambda {
    public void display();
}
```

Yeh ek functional interface hai jisme ek hi method `display()` hai.

2. **Class UseLambda:**

java Copy code

```java
class UseLambda {
    public void callLambda(MyLambda ml) {
        ml.display();
    }
}
```

Is class mein ek method `callLambda` hai jo `MyLambda` type ka object lekar uske `display` method ko call karta hai.

3. **Class Demo:**

java Copy code

```java
class Demo {
    public void method1() {
        UseLambda ul = new UseLambda();
        ul.callLambda(() -> { System.err.println("Hello"); });
    }
}
```

`Demo` class mein ek method `method1` hai jisme `UseLambda` class ka ek object banaya jata hai, aur uske `callLambda` method ko ek lambda expression ke sath call kiya jata hai. Yeh lambda expression "Hello" ko standard error stream mein print karta hai.

4. **Main class:**

java Copy code

```java
public class Main {
    public static void main(String[] args) {
        Demo d = new Demo();
        d.method1();
    }
}
```

`Main` class mein `main` method ek `Demo` class ka object banata hai aur uske `method1` ko call karta hai, jisse lambda expression ka execution hota hai.

Saransh mein:

- Yeh code dikhata hai ki kaise Java mein lambda expressions ka istemal kiya ja sakta hai.
- `MyLambda` naamak ek functional interface define ki gayi hai jisme ek hi abstract method `display` hai.
- `UseLambda` class mein ek method hai jo `MyLambda` type ka object lekar uske `display` method ko call karta hai.
- `Demo` class mein `method1` method mein ek lambda expression ke sath `UseLambda` ka object banaya jata hai, jiska `callLambda` method call hota hai.
- Lambda expression mein `display` method ki implementation "Hello" ko standard error stream mein print karne ka hai.
- `Main` class mein ek `Demo` object banaya jata hai aur uske `method1` ko call karke lambda expression ka use dikhaya jata hai.