

## INNER Class (DataStructure)



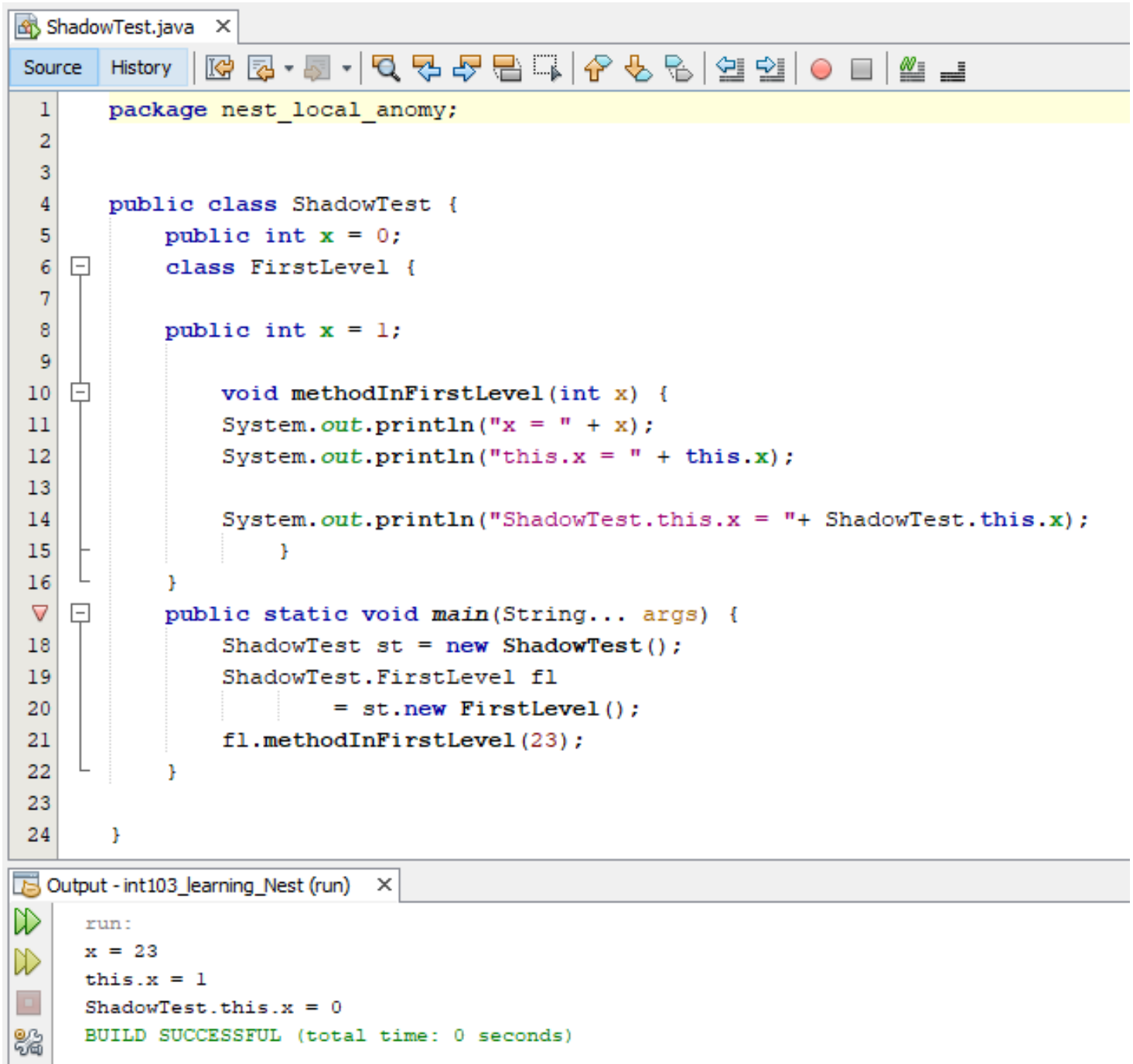
The screenshot shows an IDE with a Java file named `DataStructure.java`. The code defines a package `nest_local_anomy`, imports `java.util.Iterator`, and defines a public class `DataStructure`. Inside `DataStructure`, there is a private static final integer `SIZE` set to 15, and a private integer array `arrayOfInts` of size `SIZE`. A public constructor `DataStructure()` initializes the array with values from 0 to 14. A public void method `printEven()` creates an `EvenIterator` object and prints its values. An interface `DataStructureIterator` extends `Iterator<Integer>`. A private class `EvenIterator` implements `DataStructureIterator` and provides the `hasNext()` and `next()` methods. A public static `main` method creates a `DataStructure` object and calls `printEven()`.

```
1 package nest_local_anomy;
2 import java.util.Iterator;
3 public class DataStructure {
4     private final static int SIZE = 15;
5     private int[] arrayOfInts = new int[SIZE];
6     public DataStructure() {
7         for (int i = 0; i < SIZE; i++) {
8             arrayOfInts[i] = i;
9         }
10    }
11    public void printEven() {
12
13        DataStructureIterator iterator = this.new EvenIterator();
14        while (iterator.hasNext()) {
15            System.out.print(iterator.next() + " ");
16        }
17        System.out.println();
18    }
19    interface DataStructureIterator extends Iterator<Integer> {}
20
21    private class EvenIterator implements DataStructureIterator {
22        private int nextIndex = 0;
23        public boolean hasNext() {
24            return (nextIndex <= SIZE - 1);
25        }
26        public Integer next() {
27            Integer retValue = Integer.valueOf(arrayOfInts[nextIndex]);
28
29            nextIndex += 2;
30            return retValue;
31        }
32    }
33    public static void main(String s[]) {
34
35        DataStructure ds = new DataStructure();
36        ds.printEven();
37    }
38 }
```

The output window shows the result of running the program:

```
run:
0 2 4 6 8 10 12 14
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Inner Classes: Shadowing



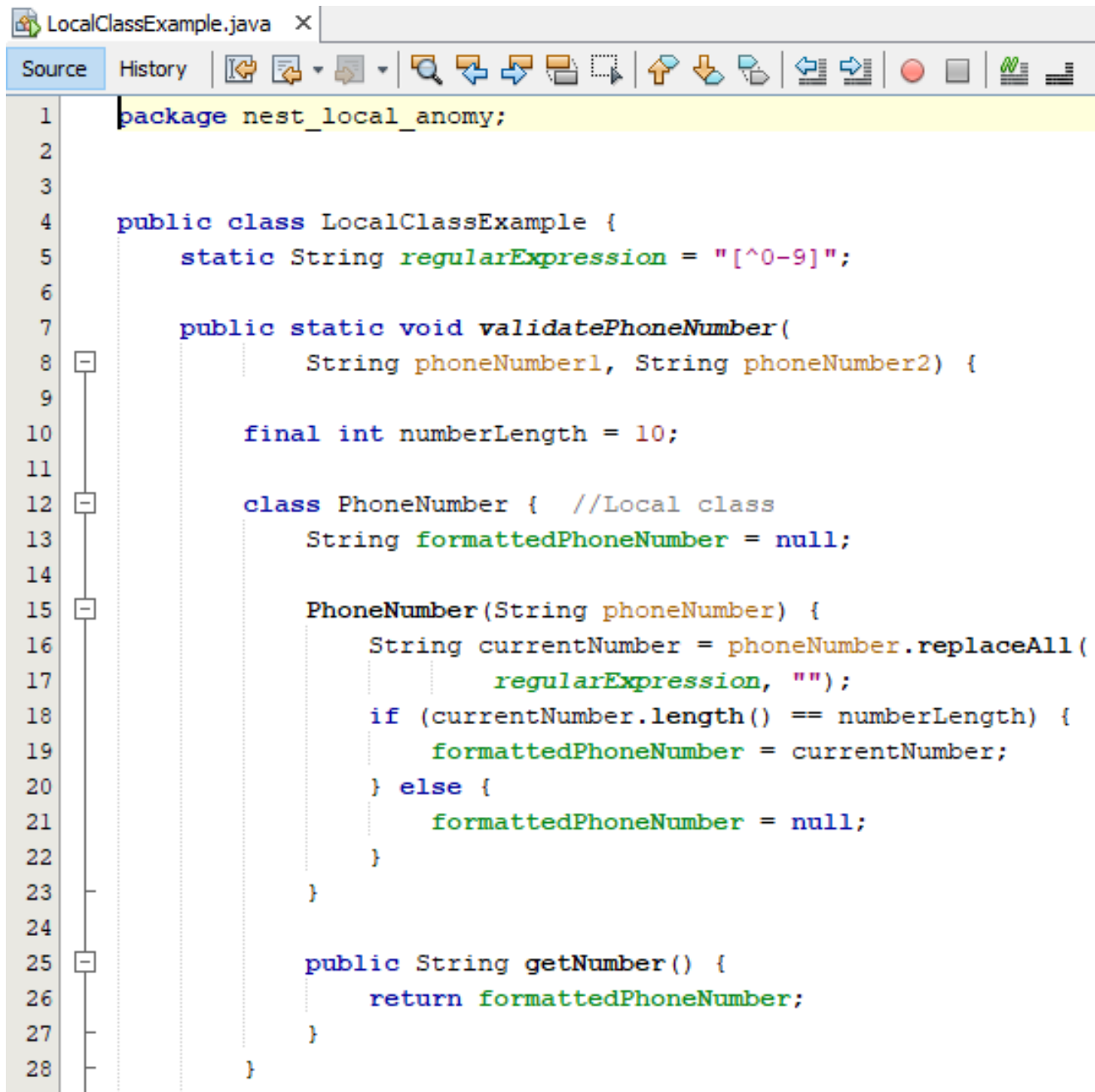
The screenshot shows an IDE window titled "ShadowTest.java" with a source code editor and an output console. The code defines a package, a public class with an inner class, and a main method that demonstrates variable shadowing.

```
1 package nest_local_anomy;
2
3
4 public class ShadowTest {
5     public int x = 0;
6     class FirstLevel {
7
8         public int x = 1;
9
10        void methodInFirstLevel(int x) {
11            System.out.println("x = " + x);
12            System.out.println("this.x = " + this.x);
13
14            System.out.println("ShadowTest.this.x = "+ ShadowTest.this.x);
15        }
16    }
17
18    public static void main(String... args) {
19        ShadowTest st = new ShadowTest();
20        ShadowTest.FirstLevel fl
21            = st.new FirstLevel();
22        fl.methodInFirstLevel(23);
23    }
24 }
```

The output console shows the following text:

```
run:
x = 23
this.x = 1
ShadowTest.this.x = 0
BUILD SUCCESSFUL (total time: 0 seconds)
```

## Local



```
1 package nest_local_anomy;
2
3
4 public class LocalClassExample {
5     static String regularExpression = "[^0-9]";
6
7     public static void validatePhoneNumber(
8         String phoneNumber1, String phoneNumber2) {
9
10        final int numberLength = 10;
11
12        class PhoneNumber { //Local class
13            String formattedPhoneNumber = null;
14
15            PhoneNumber(String phoneNumber) {
16                String currentNumber = phoneNumber.replaceAll(
17                    regularExpression, "");
18                if (currentNumber.length() == numberLength) {
19                    formattedPhoneNumber = currentNumber;
20                } else {
21                    formattedPhoneNumber = null;
22                }
23            }
24
25            public String getNumber() {
26                return formattedPhoneNumber;
27            }
28        }
```

## Local 2

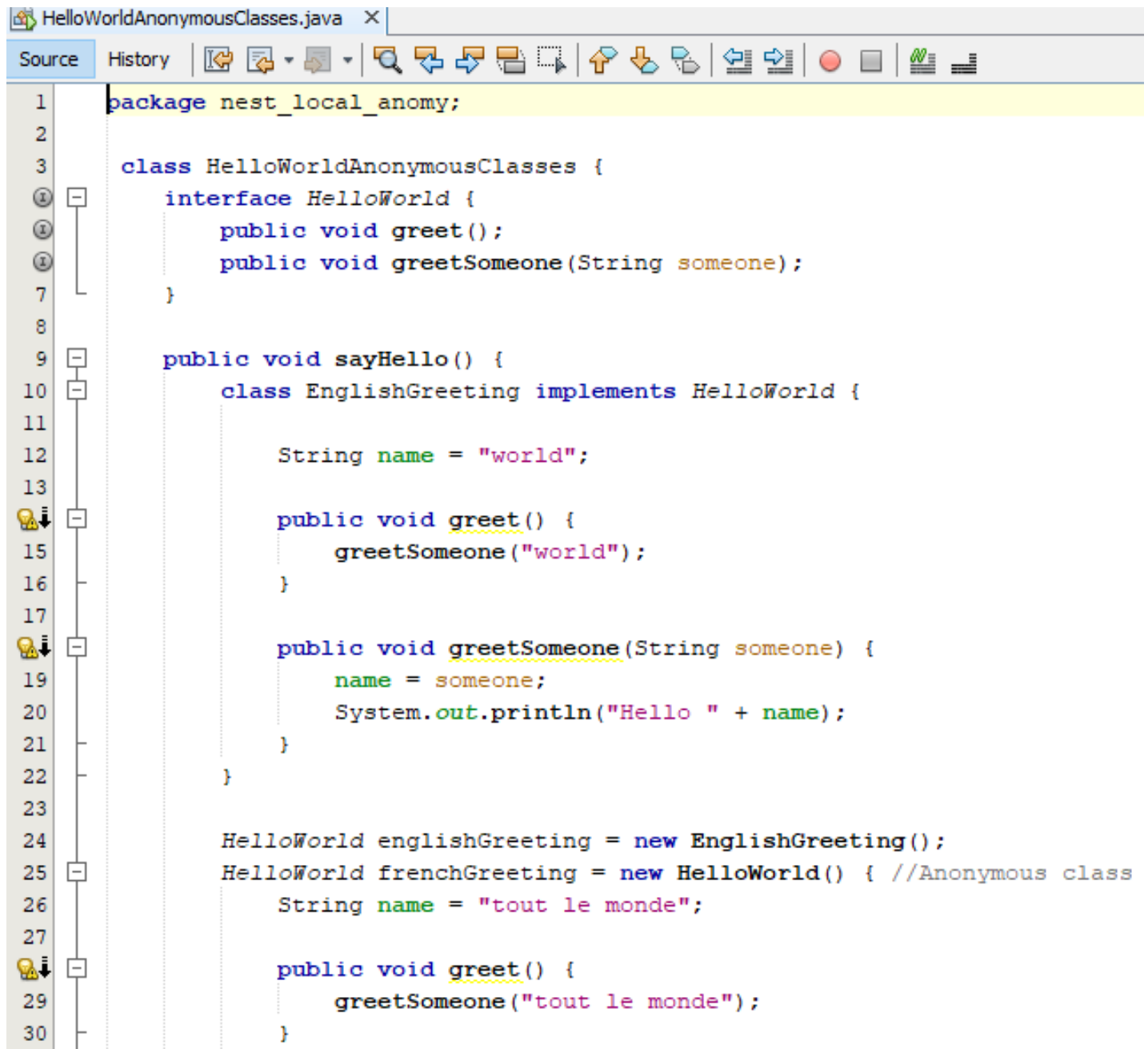
```
30     PhoneNumber myNumber1 = new PhoneNumber(phoneNumber1);
31     PhoneNumber myNumber2 = new PhoneNumber(phoneNumber2);
32
33     if (myNumber1.getNumber() == null) {
34         System.out.println("First number is invalid");
35     } else {
36         System.out.println("First number is " + myNumber1.getNumber());
37     }
38     if (myNumber2.getNumber() == null) {
39         System.out.println("Second number is invalid");
40     } else {
41         System.out.println("Second number is " + myNumber2.getNumber());
42     }
43 }
44
45 public static void main(String... args) {
46     validatePhoneNumber("123-456-7890", "456-7890");
47 }
48
49 }
50
```

Output - int103\_learning\_Nest (run) X



run:  
First number is 1234567890  
Second number is invalid  
BUILD SUCCESSFUL (total time: 0 seconds)

## Anonymous



```
1 package nest_local_anomy;
2
3 class HelloWorldAnonymousClasses {
4     interface HelloWorld {
5         public void greet();
6         public void greetSomeone(String someone);
7     }
8
9     public void sayHello() {
10         class EnglishGreeting implements HelloWorld {
11             String name = "world";
12
13             public void greet() {
14                 greetSomeone("world");
15             }
16
17             public void greetSomeone(String someone) {
18                 name = someone;
19                 System.out.println("Hello " + name);
20             }
21         }
22
23         HelloWorld englishGreeting = new EnglishGreeting();
24         HelloWorld frenchGreeting = new HelloWorld() { //Anonymous class
25             String name = "tout le monde";
26
27             public void greet() {
28                 greetSomeone("tout le monde");
29             }
30         }
```

## Anonymous 2

```
33      public void greetSomeone(String someone) {
34          name = someone;
35          System.out.println("Salut " + name);
36      }
37
38      };
39
40      englishGreeting.greet();
41      frenchGreeting.greetSomeone("Fred");
42  }
43
44  public static void main(String... args) {
45      HelloWorldAnonymousClasses myApp
46          = new HelloWorldAnonymousClasses();
47      myApp.sayHello();
48  }
```

Output - int103\_learning\_Nest (run) X

```
run:
First number is 1234567890
Second number is invalid
BUILD SUCCESSFUL (total time: 0 seconds)
```