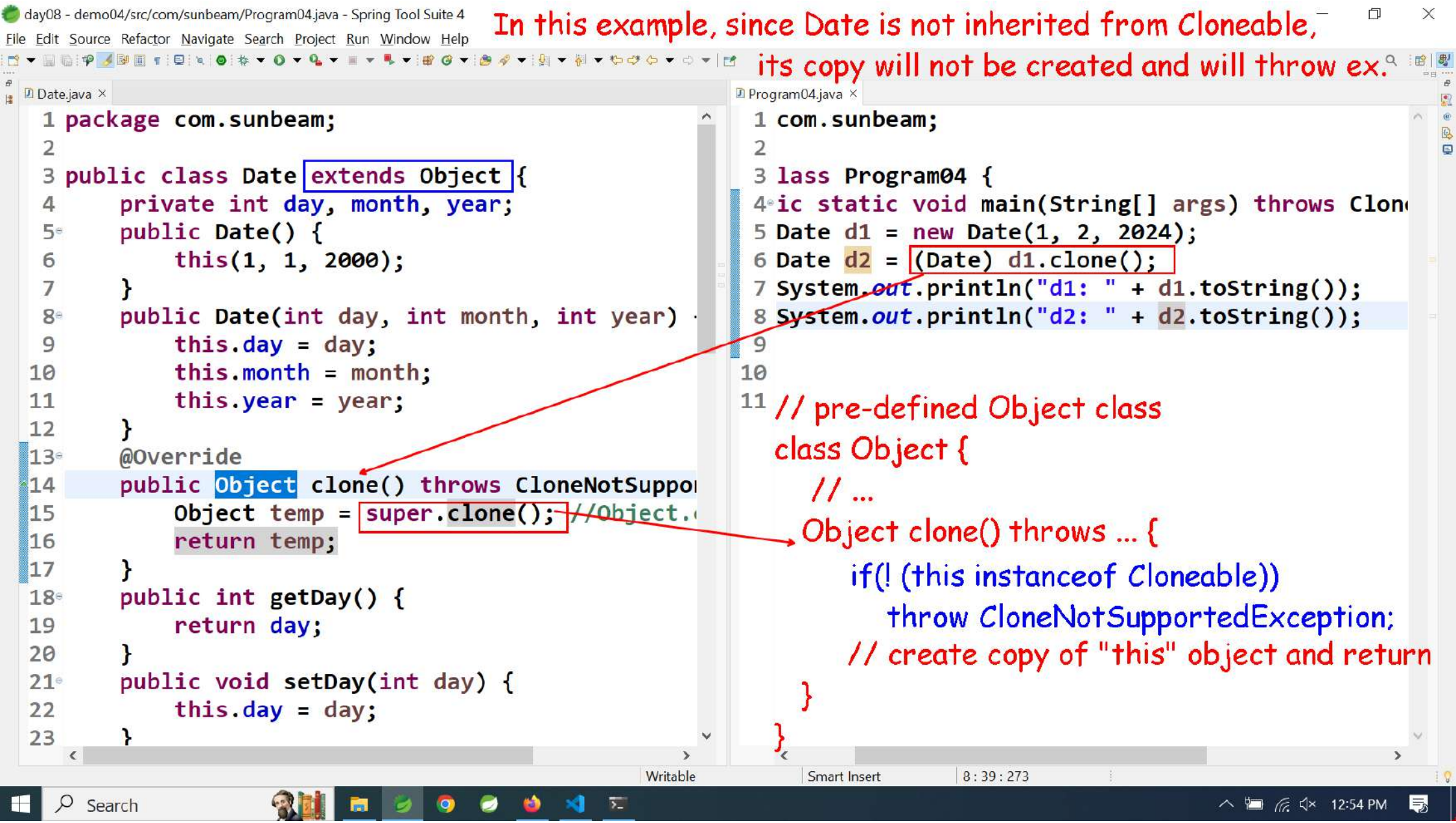
-instance-of
    - checks the reference of same type or sub-class
    - instance-of is used to before doing downcasting
    - to call non-overrided methods we need to have reference of that specific class
- final method
    - if implementation of method is logically 100% complete we can declare method as final
- final class
    - if implementation of class is 100% complete we declare class as final , eventually all the methods of final class becomes final
- equals()
    - to compare the state of object we override the equals method
        - comparison of same type
        - null comparison
        - incompatible types (instance-of) -- ClassCastException
- interface
    - Fragile base class problem - (changes made in super class -- recompile all the sub-class)
    - interface - standards / rules / specification |

- interface
  - interface contains only method declaration and field declaration
  - field -(public static final) , method (public abstract)
  - multiple interface inheritance is allowed in java (interface only contains declaration)

- abstract method
  - implementation of method is incomplete method as abstract
  - abstract method cannot be private ,static , final
  - if method is abstract we need to declare class as abstract
  - abstract method are forced to be implemented in subclass or else mark sublcass as abstract
  - abstract method are forced to be implemented in subclass because sub-class should have corresponding behaviour

- abstract class
  - if implementation of class is logically incomplete - declare class as abstract
  - abstract class may contain zero or more abstract method
  - method is abstract -- declare class as abstract
  - abstract class can have fields , methods and constructors (we can create refereene) I


day08 - demo04/src/com/sunbeam/Program04.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
In this example, since Date is not inherited from Cloneable,
its copy will not be created and will throw ex.
Date.java
1 package com.sunbeam;
2
3 public class Date extends Object {
4 private int day, month, year;
5 public Date() {
6 this(1, 1, 2000);
7 }
8 public Date(int day, int month, int year)
9 this.day = day;
10 this.month = month;
11 this.year = year;
12 }
13 @Override
14 public Object clone() throws CloneNotSuppo
15 Object temp = super.clone(); //Object.
16 return temp;
17 }
18 public int getDay() {
19 return day;
20 }
21 public void setDay(int day) {
22 this.day = day;
23 }
Program04.java
1 com.sunbeam;
2
3 lass Program04 {
4 ic static void main(String[] args) throws Clon
5 Date d1 = new Date(1, 2, 2024);
6 Date d2 = (Date) d1.clone();
7 System.out.println("d1: " + d1.toString());
8 System.out.println("d2: " + d2.toString());
9
10
11 // pre-defined Object class
class Object {
// ...
Object clone() throws ... {
if(! (this instanceof Cloneable))
throw CloneNotSupportedException;
// create copy of "this" object and return
}
}
Writable
Smart Insert
8 : 39 : 273
Search
12:54 PM

Deep copy - Copy of outer object as well as inner objects (referred by the fields in outer object). Now both objects have different memory locations. Changes in one will not affect other object.

day09 - demo01/src/com/sunbeam/Person.java - Spring Tool Suite 4

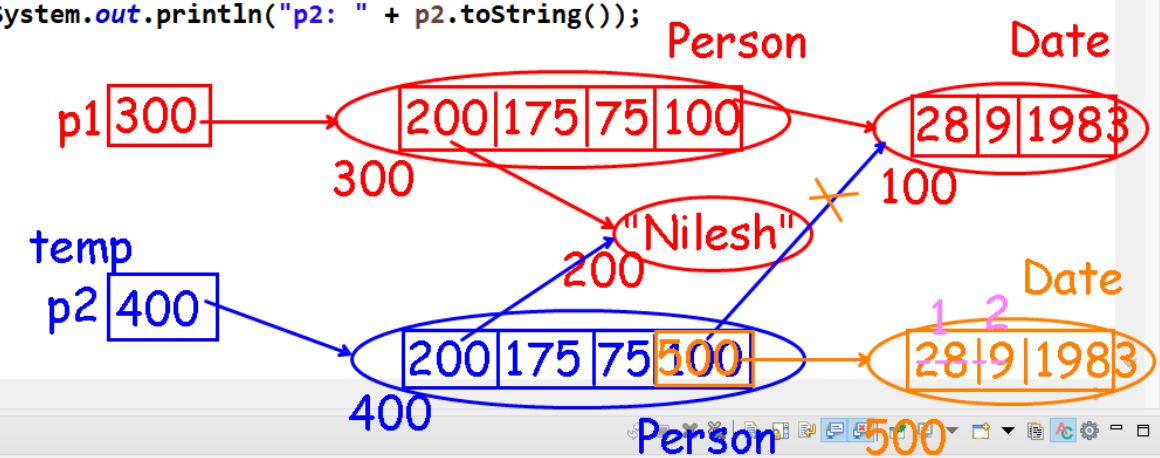
File Edit Source Refactor Navigate Search Project Run Window Help

Date.java | Person.java

```
3  public class Person implements Cloneable {
4      private String name;
5      private int height, weight;
6      private Date birth;
7
8      public Person() {
11     public Person(String name, int height, int wei
17     public int getHeight() {
20     public void setHeight(int height) {
23     public int getWeight() {
26     public void setWeight(int weight) {
29     public String getName() {
32     public void setName(String name) {
35     public Date getBirth() {
38     public void setBirth(Date birth) {
42     public String toString() {
45
46     @Override
47     protected Object clone() throws CloneNotSuppor
48         Person temp = (Person) super.clone();
49         temp.birth = (Date) this.birth.clone();
50         return temp;
```

Program01.java

```
1  package com.sunbeam;
2
3  public class Program01 {
4      public static void main(String[] args) throws CloneNotSupportedException {
5          Person p1 = new Person("Nilesh", 175, 75, new Date(28, 9, 1983));
6          Person p2 = (Person) p1.clone();
7          System.out.println("p1: " + p1.toString());
8          System.out.println("p2: " + p2.toString());
9          p2.getBirth().setDay(1);
10         p2.getBirth().setMonth(2);
11         System.out.println("p1: " + p1.toString());
12         System.out.println("p2: " + p2.toString());
13     }
14 }
15
```



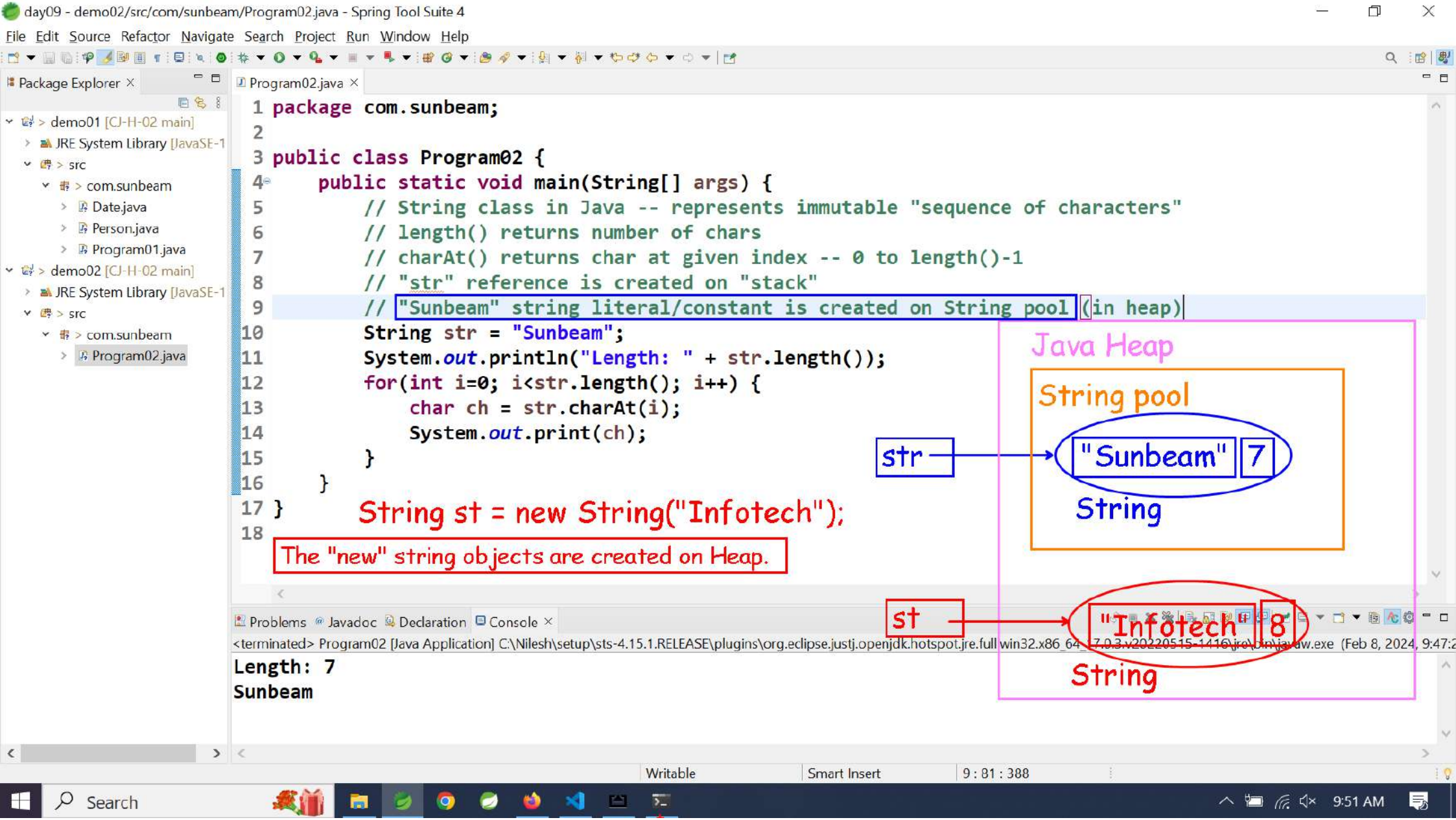
Problems | Javadoc | Declaration | Console

<terminated> Program01 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 9:15:17 AM – 9:15:17 AM) [pid: 2940]

```
p1: Person [name=Nilesh, height=175, weight=75, birth=28-9-1983]
p2: Person [name=Nilesh, height=175, weight=75, birth=28-9-1983]
p1: Person [name=Nilesh, height=175, weight=75, birth=1-2-1983]
p2: Person [name=Nilesh, height=175, weight=75, birth=1-2-1983]
```

Writable | Smart Insert | 45 : 5 : 945

Search 9:26 AM


day09 - demo02/src/com/sunbeam/Program02.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
demo01 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Date.java
Person.java
Program01.java
demo02 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program02.java
Program02.java
1 package com.sunbeam;
2
3 public class Program02 {
4 public static void main(String[] args) {
5 // String class in Java -- represents immutable "sequence of characters"
6 // length() returns number of chars
7 // charAt() returns char at given index -- 0 to length()-1
8 // "str" reference is created on "stack"
9 // "Sunbeam" string literal/constant is created on String pool (in heap)
10 String str = "Sunbeam";
11 System.out.println("Length: " + str.length());
12 for(int i=0; i<str.length(); i++) {
13 char ch = str.charAt(i);
14 System.out.print(ch);
15 }
16 }
17 }
18
String st = new String("Infotech");
The "new" string objects are created on Heap.
Java Heap
String pool
str
"Sunbeam"
7
String
st
"Infotech"
8
String
Problems Javadoc Declaration Console
<terminated> Program02 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 9:47:2
Length: 7
Sunbeam
Writable
Smart Insert
9 : 81 : 388
Search
9:51 AM

File Edit Selection View Go ··· private

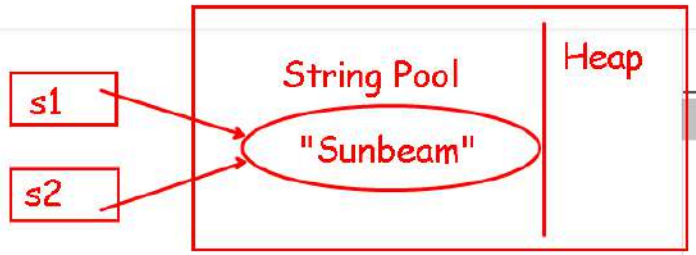
day09.md | day08.md | classwork.md U | Untitled-1

day09.md > # Core Java > ## Java Strings > ### String literals

String objects vs String literals

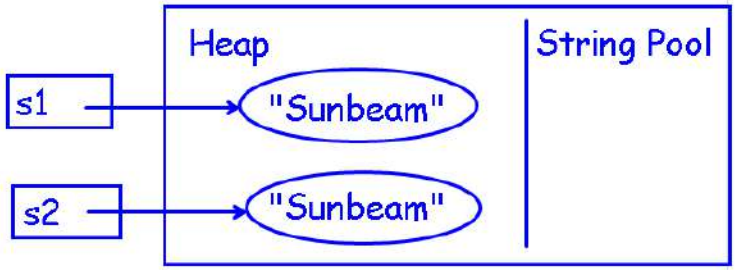
* Example 01:

```Java
String s1 = "Sunbeam";
String s2 = "Sunbeam";
System.out.println(s1 == s2);        // ???true
System.out.println(s1.equals(s2));   // ???true
```



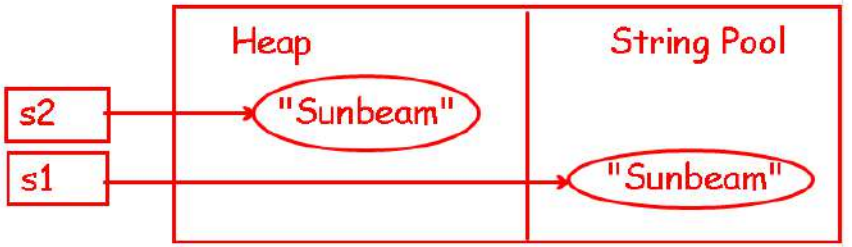
* Example 02:

```Java
String s1 = new String("Sunbeam");
String s2 = new String("Sunbeam");
System.out.println(s1 == s2);        // ??? false
System.out.println(s1.equals(s2));   // ??? true
```



* Example 03:

```Java
String s1 = "Sunbeam";
String s2 = new String("Sunbeam");
System.out.println(s1 == s2);        // ??? false
System.out.println(s1.equals(s2));   // ??? true
```



* Example 04:

main* ⊗0 △0 0 | Ln 35, Col 58 (55 selected) | Spaces: 4 | UTF-8 | CRLF | Markdown

Search 10:28 AM

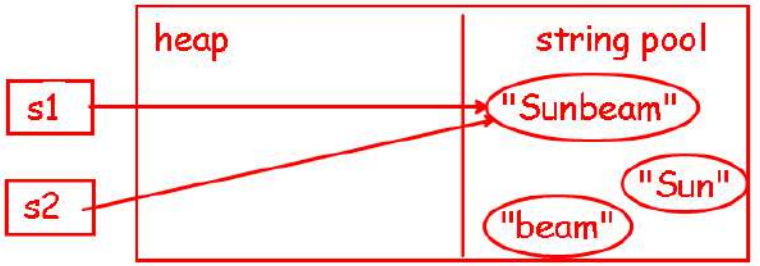
File Edit Selection View Go ··· private

day09.md | day08.md | classwork.md U | Untitled-1

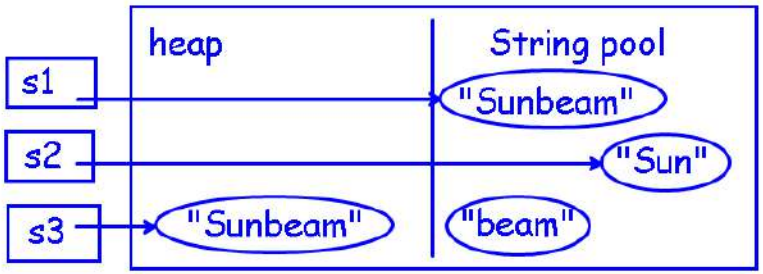
day09.md > # Core Java > ## Java Strings > ### String literals

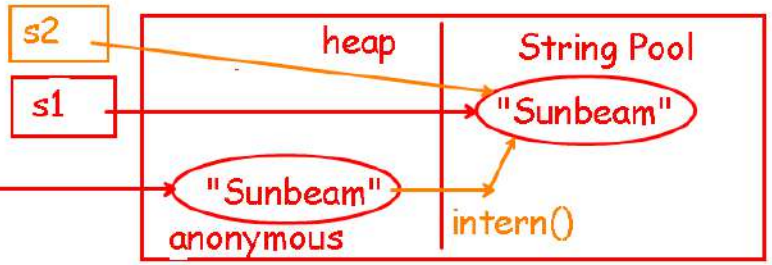
````
59  ```
60  * Example 04:
61  ```Java
62  String s1 = "Sunbeam";
63  String s2 = "Sun" + "beam";
64  System.out.println(s1 == s2);          // ??? true
65  System.out.println(s1.equals(s2));     // ??? true
66  ```
67  * Example 05:
68  ```Java
69  String s1 = "Sunbeam";
70  String s2 = "Sun";
71  String s3 = s2 + "beam";
72  System.out.println(s1 == s3);          // ??? false
73  System.out.println(s1.equals(s3));     // ??? true
74  ```
75  * Example 06:
76  ```Java
77  String s1 = "Sunbeam";
78  String s2 = new String("Sunbeam").intern();
79  System.out.println(s1 == s2);          // ??? true
80  System.out.println(s1.equals(s2));     // ??? true
81  ```
````

evaluated by compiler
"Sunbeam" is already in pool.



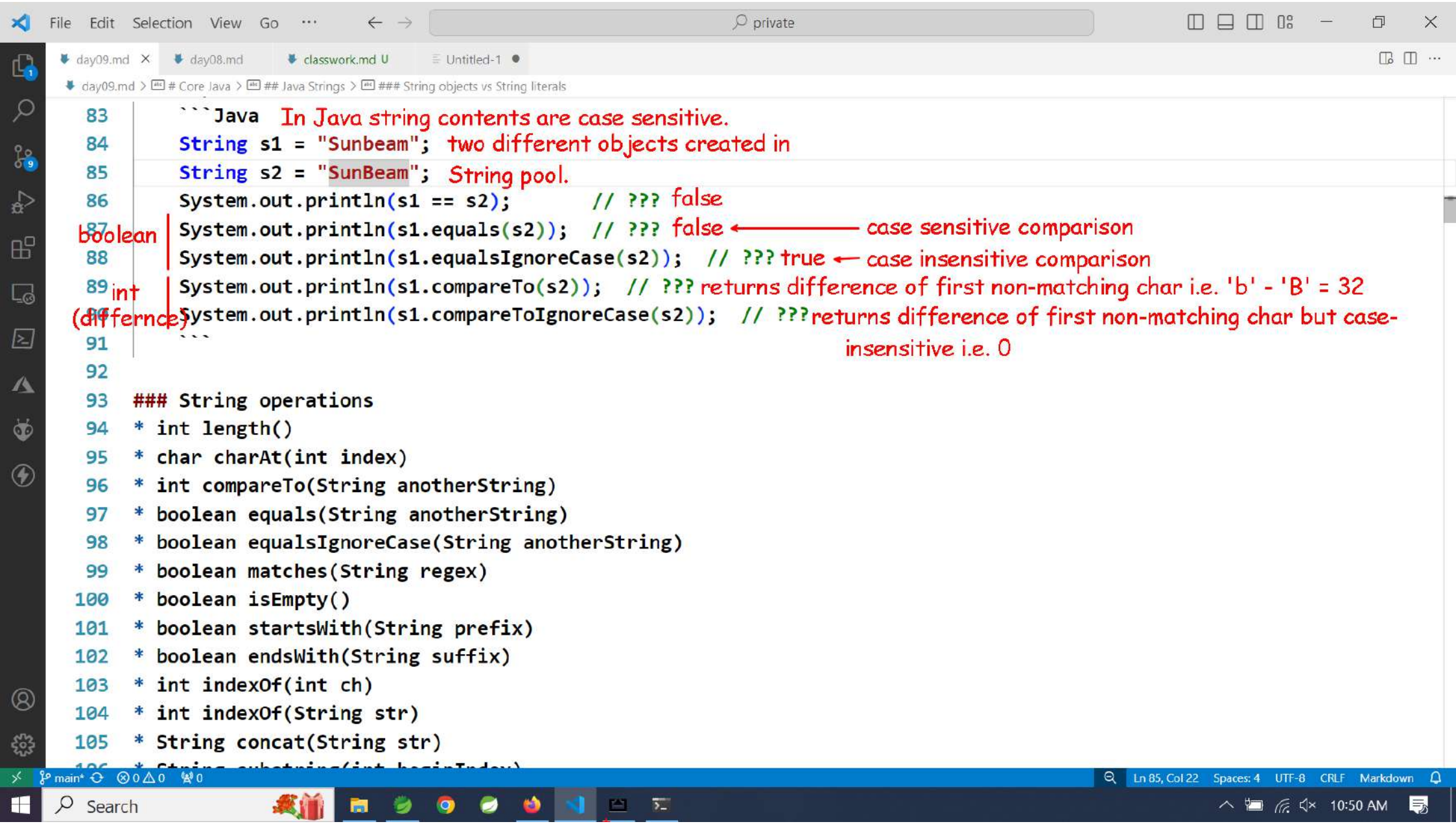
compiler generate byte code
evaluated by jvm at runtime
new obj in heap

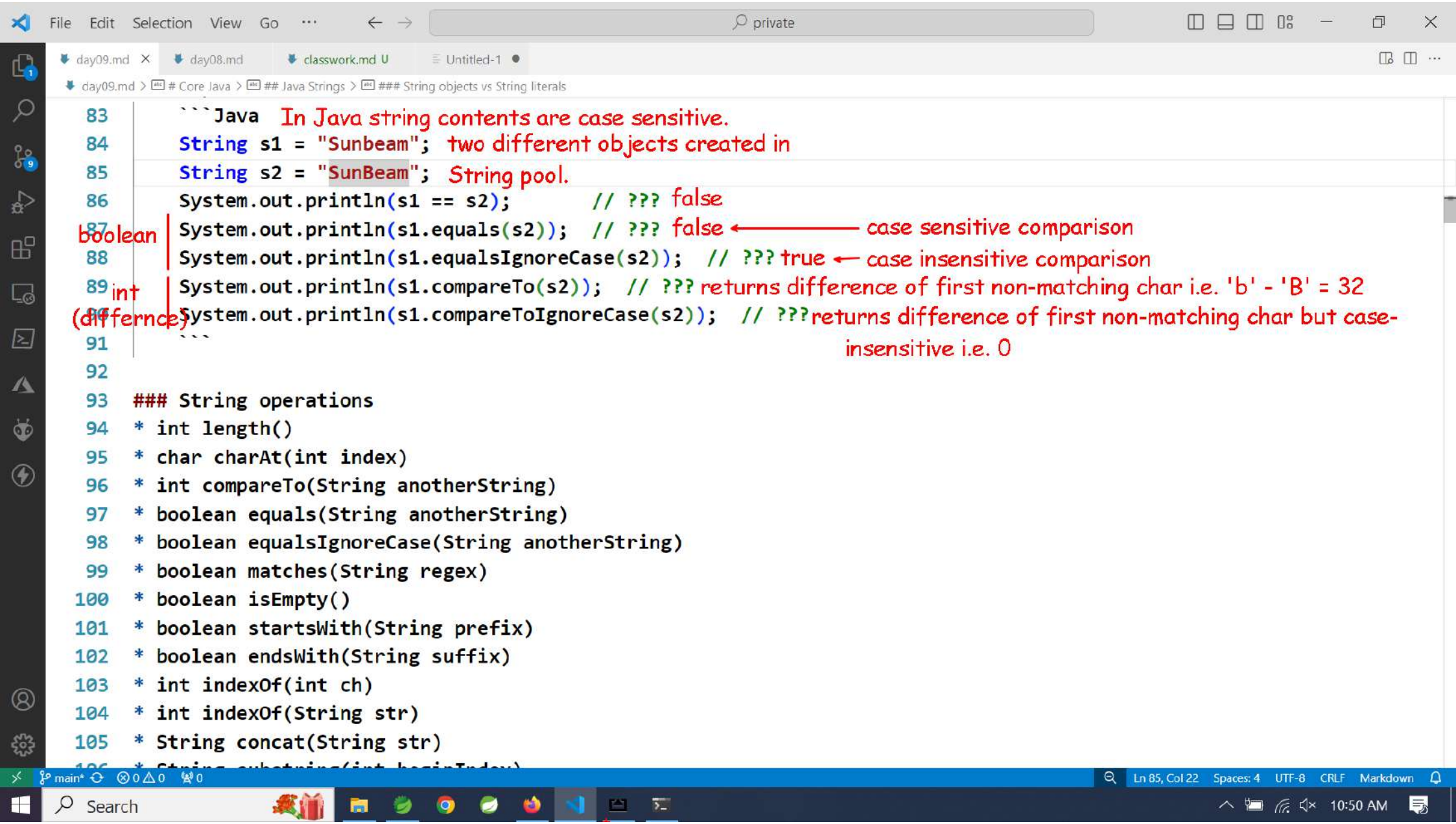




main* ⊗0 △0 0 | Ln 35, Col 58 (55 selected) Spaces: 4 UTF-8 CRLF Markdown

Search 10:36 AM


File Edit Selection View Go
private
day09.md
day08.md
classwork.md U
Untitled-1
day09.md > # Core Java > ## Java Strings > ### String objects vs String literals
```Java
String s1 = "Sunbeam";
String s2 = "SunBeam";
System.out.println(s1 == s2);
System.out.println(s1.equals(s2));
System.out.println(s1.equalsIgnoreCase(s2));
System.out.println(s1.compareTo(s2));
System.out.println(s1.compareToIgnoreCase(s2));
```
In Java string contents are case sensitive.
two different objects created in String pool.
// ??? false
// ??? false
case sensitive comparison
// ??? true
case insensitive comparison
boolean
int (difference)
// ??? returns difference of first non-matching char i.e. 'b' - 'B' = 32
// ??? returns difference of first non-matching char but case-insensitive i.e. 0
String operations
* int length()
* char charAt(int index)
* int compareTo(String anotherString)
* boolean equals(String anotherString)
* boolean equalsIgnoreCase(String anotherString)
* boolean matches(String regex)
* boolean isEmpty()
* boolean startsWith(String prefix)
* boolean endsWith(String suffix)
* int indexOf(int ch)
* int indexOf(String str)
* String concat(String str)
main*
Ln 85, Col 22
Spaces: 4
UTF-8
CRLF
Markdown
Search
10:50 AM


File Edit Selection View Go
private
day09.md
day08.md
classwork.md U
Untitled-1
day09.md > # Core Java > ## Java Strings > ### String objects vs String literals
```Java
String s1 = "Sunbeam";
String s2 = "SunBeam";
System.out.println(s1 == s2);
System.out.println(s1.equals(s2));
System.out.println(s1.equalsIgnoreCase(s2));
System.out.println(s1.compareTo(s2));
System.out.println(s1.compareToIgnoreCase(s2));
```
In Java string contents are case sensitive.
two different objects created in String pool.
// ??? false
// ??? false
case sensitive comparison
// ??? true
case insensitive comparison
boolean
int (difference)
// ??? returns difference of first non-matching char i.e. 'b' - 'B' = 32
// ??? returns difference of first non-matching char but case-insensitive i.e. 0
String operations
* int length()
* char charAt(int index)
* int compareTo(String anotherString)
* boolean equals(String anotherString)
* boolean equalsIgnoreCase(String anotherString)
* boolean matches(String regex)
* boolean isEmpty()
* boolean startsWith(String prefix)
* boolean endsWith(String suffix)
* int indexOf(int ch)
* int indexOf(String str)
* String concat(String str)
main*
Ln 85, Col 22
Spaces: 4
UTF-8
CRLF
Markdown
Search
10:50 AM

day09 - demo03/src/com/sunbeam/Program03.java - Spring Tool Suite 4

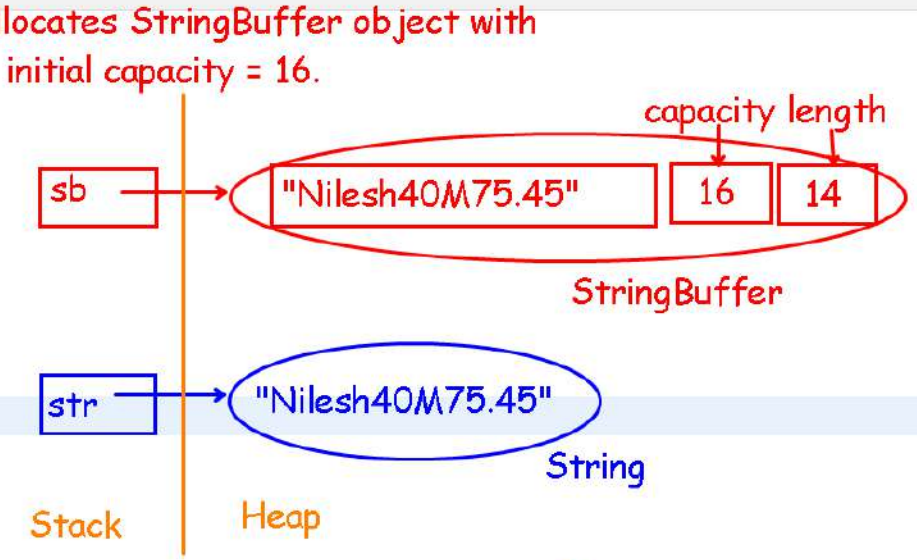
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- demo01 [CJ-H-02 main]
  - JRE System Library [JavaSE-1
  - src
    - com.sunbeam
      - Date.java
      - Person.java
      - Program01.java
- demo02 [CJ-H-02 main]
  - JRE System Library [JavaSE-1
  - src
    - com.sunbeam
      - Program02.java
- demo03 [CJ-H-02 main]
  - JRE System Library [JavaSE-1
  - src
    - com.sunbeam
      - Program03.java

Program02.java | Program03.java

```
1  package com.sunbeam;
2
3  public class Program03 {
4      public static void main(String[] args) {
5          StringBuffer sb = new StringBuffer();
6          sb.append("Nilesh"); // append(String)
7          sb.append(40); // append(int)
8          sb.append('M'); // append(char)
9          sb.append(75.45); // append(double)
10         String str = sb.toString();
11         System.out.println(str);
12     }
13
14     /*
15     public static void main(String[] args) {
16         // capacity is size of internal char array
17         // length is number of chars stored in that array
18         StringBuffer sb = new StringBuffer();
19         System.out.println("Capacity: " + sb.capacity() + ", Length: " + sb.length()); // Capacity: ?
```



Problems Javadoc Declaration Console

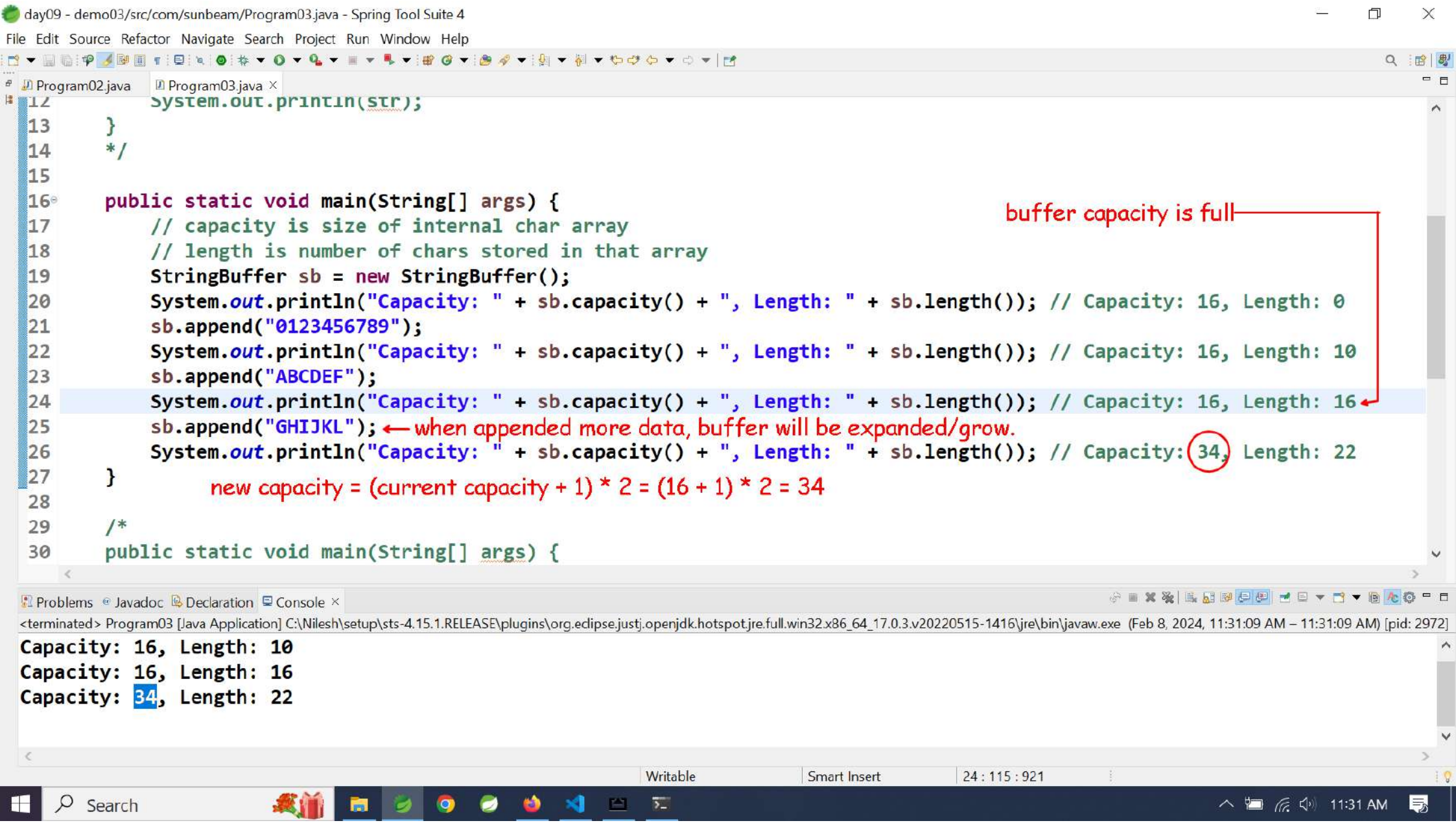
<terminated> Program03 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 11:23

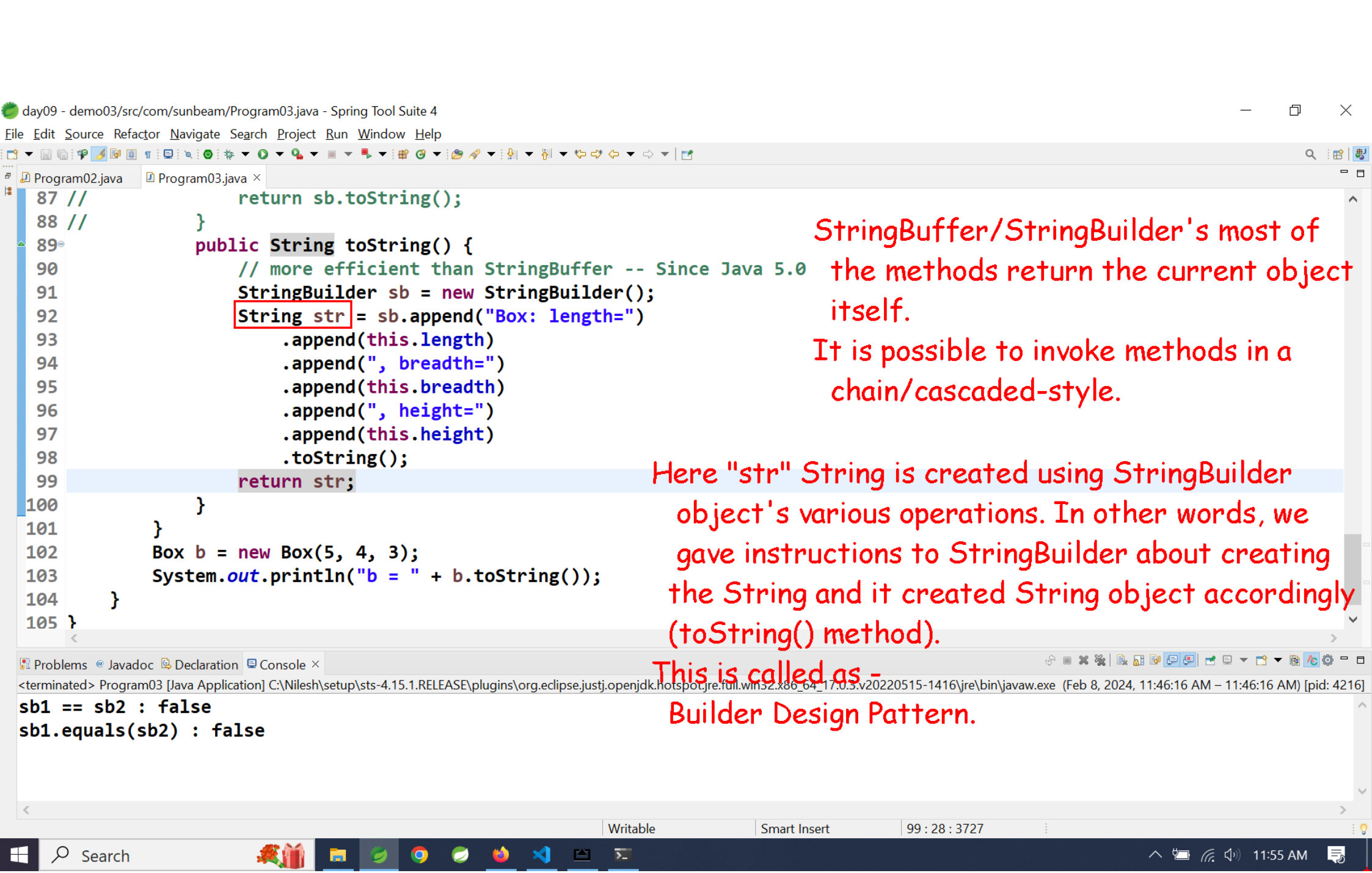
```
Nilesh40M75.45
```

Writable | Smart Insert | 11 : 31 : 338

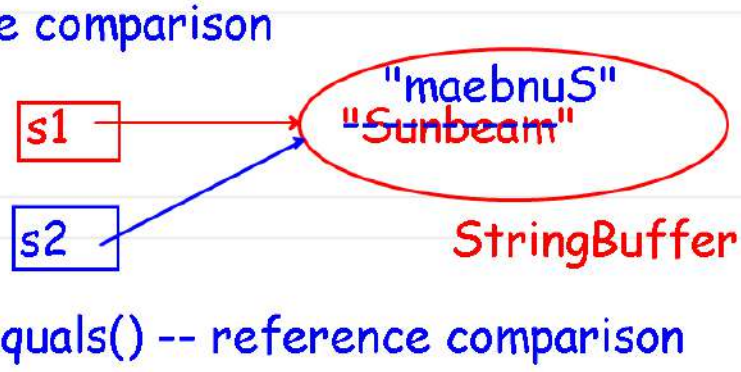
Search

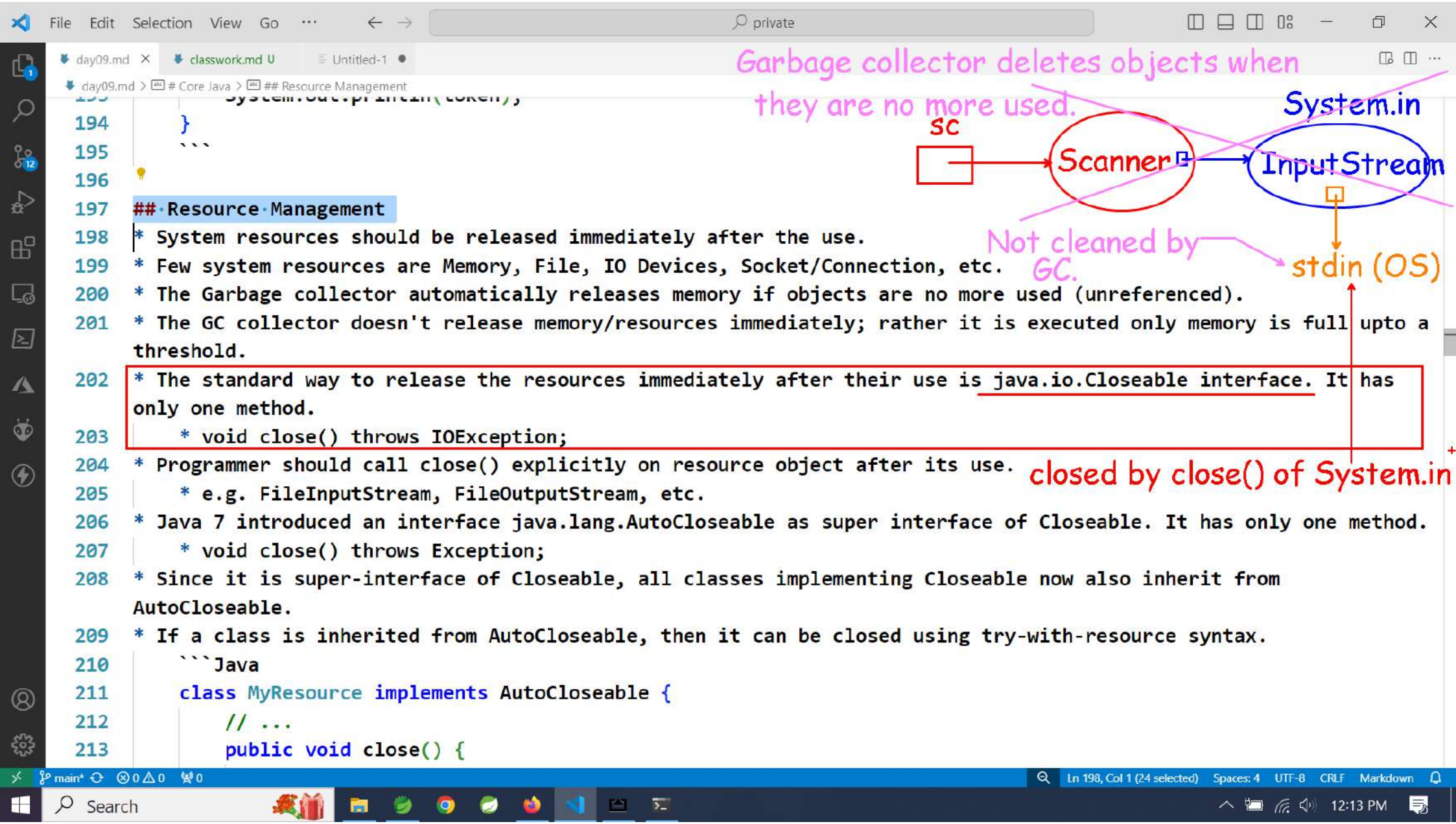
11:23 AM

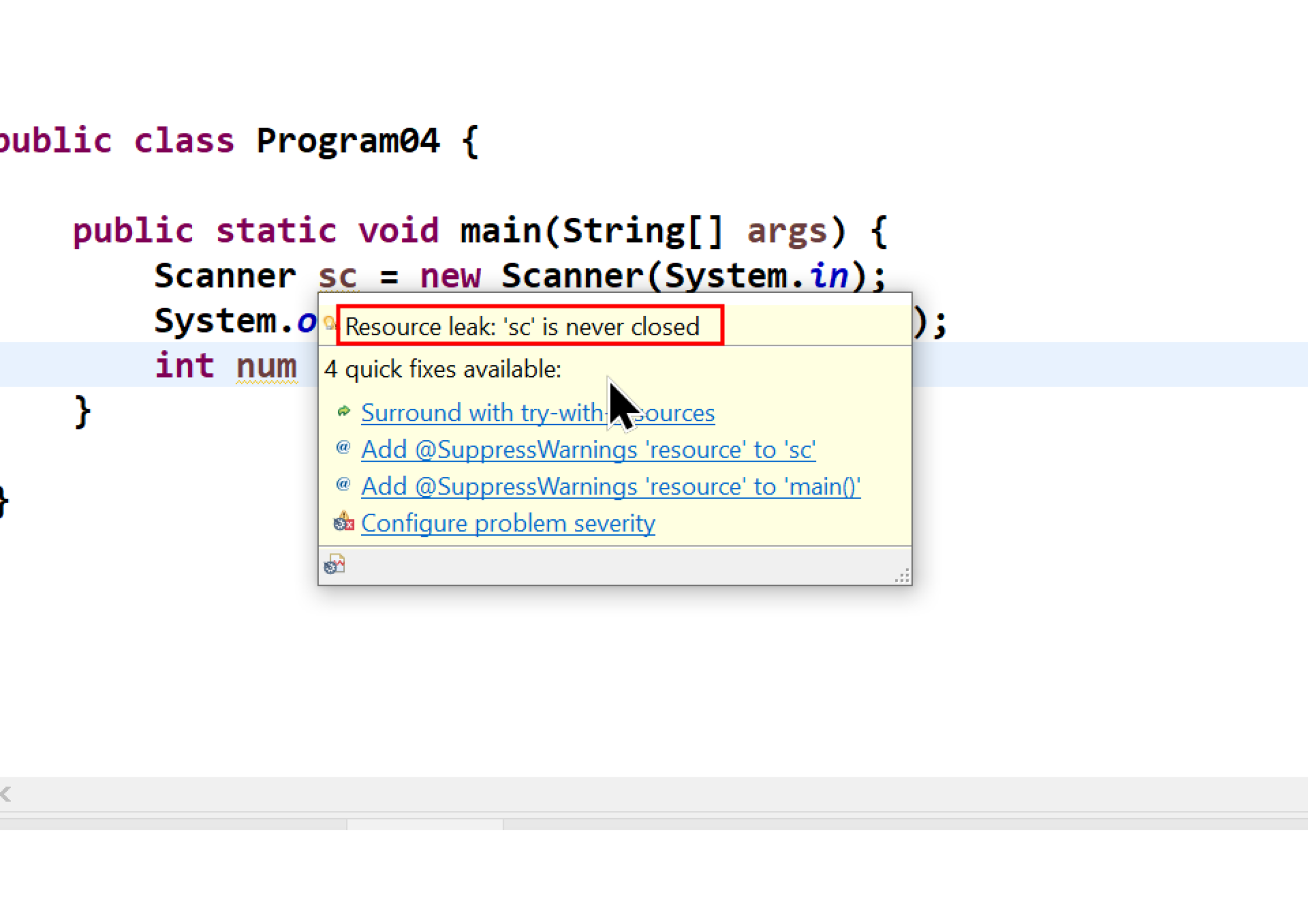

day09 - demo03/src/com/sunbeam/Program03.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Program02.java
Program03.java
System.out.println(str);
}
*/
public static void main(String[] args) {
// capacity is size of internal char array
// length is number of chars stored in that array
StringBuffer sb = new StringBuffer();
System.out.println("Capacity: " + sb.capacity() + ", Length: " + sb.length()); // Capacity: 16, Length: 0
sb.append("0123456789");
System.out.println("Capacity: " + sb.capacity() + ", Length: " + sb.length()); // Capacity: 16, Length: 10
sb.append("ABCDEF");
System.out.println("Capacity: " + sb.capacity() + ", Length: " + sb.length()); // Capacity: 16, Length: 16
sb.append("GHIJKL");
System.out.println("Capacity: " + sb.capacity() + ", Length: " + sb.length()); // Capacity: 34, Length: 22
}
/*
public static void main(String[] args) {
buffer capacity is full
when appended more data, buffer will be expanded/grow.
new capacity = (current capacity + 1) * 2 = (16 + 1) * 2 = 34
Problems Javadoc Declaration Console
<terminated> Program03 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 11:31:09 AM – 11:31:09 AM) [pid: 2972]
Capacity: 16, Length: 10
Capacity: 16, Length: 16
Capacity: 34, Length: 22
Writable
Smart Insert
24 : 115 : 921
Search
11:31 AM

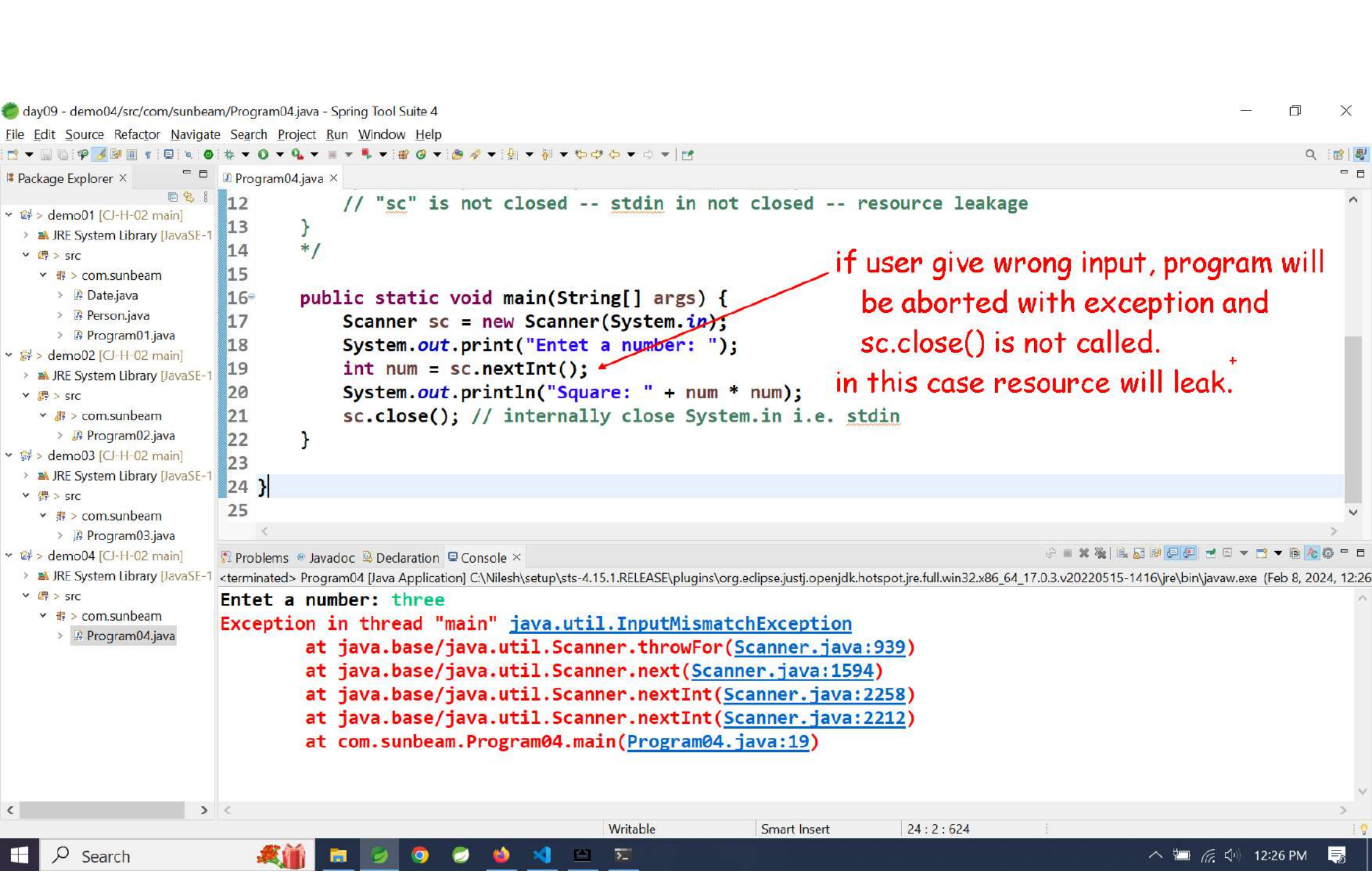

day09 - demo03/src/com/sunbeam/Program03.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Program02.java
Program03.java
87 //   return sb.toString();
88 //   }
89   public String toString() {
90   // more efficient than StringBuffer -- Since Java 5.0
91   StringBuilder sb = new StringBuilder();
92   String str = sb.append("Box: length=")
93   .append(this.length)
94   .append(", breadth=")
95   .append(this.breadth)
96   .append(", height=")
97   .append(this.height)
98   .toString();
99   return str;
100   }
101   }
102   Box b = new Box(5, 4, 3);
103   System.out.println("b = " + b.toString());
104 }
105 }
StringBuffer/StringBuilder's most of the methods return the current object itself.
It is possible to invoke methods in a chain/cascaded-style.
Here "str" String is created using StringBuilder object's various operations. In other words, we gave instructions to StringBuilder about creating the String and it created String object accordingly (toString() method).
This is called as - Builder Design Pattern.
Problems Javadoc Declaration Console
<terminated> Program03 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 11:46:16 AM – 11:46:16 AM) [pid: 4216]
sb1 == sb2 : false
sb1.equals(sb2) : false
Writable
Smart Insert
99 : 28 : 3727
Search
11:55 AM

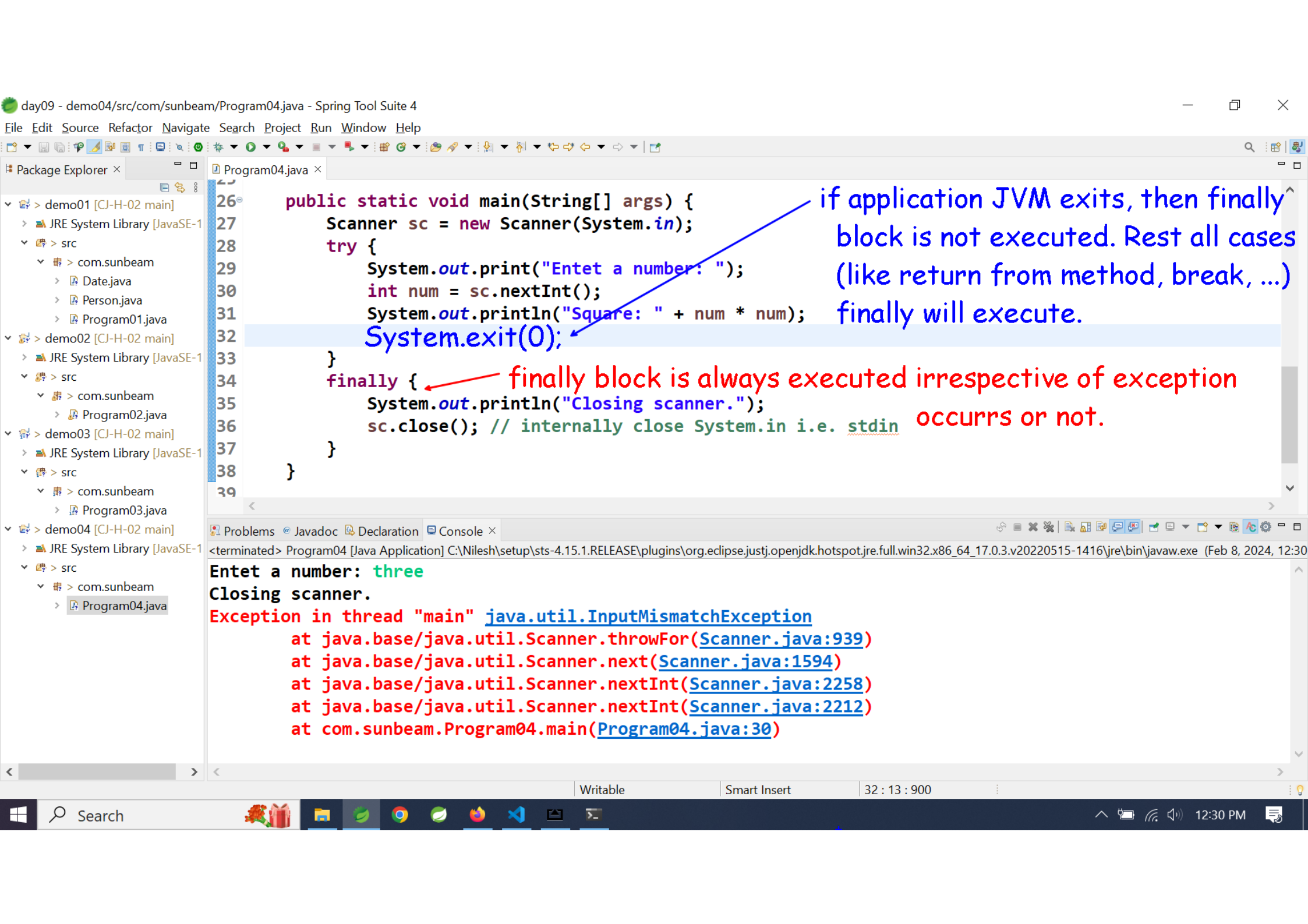
```
* Example 04:
    ```Java
    StringBuffer s1 = new StringBuffer("Sunbeam");
    StringBuffer s2 = s1.reverse();
    System.out.println(s1 == s2);       // true
    System.out.println(s1.equals(s2));  // true
    ```
* Example 05:
    ```Java
    StringBuilder s1 = new StringBuilder("Sunbeam");
    StringBuilder s2 = new StringBuilder("Sunbeam");
    System.out.println(s1 == s2);       // ???
    System.out.println(s1.equals(s2));  // ???
    ```
* Example 06:
    ```Java
    StringBuffer s = new StringBuffer();
    System.out.println("Capacity: " + s.capacity() + ", Length: " + s.length()); // 16, 0
    s.append("1234567890");
    System.out.println("Capacity: " + s.capacity() + ", Length: " + s.length()); // 16, 10
    s.append("ABCDEFGHIJKLMNOPQRSTUV");
    System.out.println("Capacity: " + s.capacity() + ", Length: " + s.length()); // 34, 32
    ```
```




File Edit Selection View Go
private
day09.md
classwork.md U
Untitled-1
day09.md > # Core Java > ## Resource Management
Garbage collector deletes objects when they are no more used.
sc
Scanner
System.in
InputStream
Not cleaned by GC.
stdin (OS)
closed by close() of System.in
194 }
195 ```
196
197 ## Resource Management
198 * System resources should be released immediately after the use.
199 * Few system resources are Memory, File, IO Devices, Socket/Connection, etc.
200 * The Garbage collector automatically releases memory if objects are no more used (unreferenced).
201 * The GC collector doesn't release memory/resources immediately; rather it is executed only memory is full upto a threshold.
202 * The standard way to release the resources immediately after their use is java.io.Closeable interface. It has only one method.
203 * void close() throws IOException;
204 * Programmer should call close() explicitly on resource object after its use.
205 * e.g. FileInputStream, FileOutputStream, etc.
206 * Java 7 introduced an interface java.lang.AutoCloseable as super interface of Closeable. It has only one method.
207 * void close() throws Exception;
208 * Since it is super-interface of Closeable, all classes implementing Closeable now also inherit from AutoCloseable.
209 * If a class is inherited from AutoCloseable, then it can be closed using try-with-resource syntax.
210 ```Java
211 class MyResource implements AutoCloseable {
212 // ...
213 public void close() {
main* 0 0 0
Ln 198, Col 1 (24 selected) Spaces: 4 UTF-8 CRLF Markdown
Search
12:13 PM


public class Program04 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.o
);
int num
}
}
Resource leak: 'sc' is never closed
4 quick fixes available:
Surround with try-with-resources
Add @SuppressWarnings 'resource' to 'sc'
Add @SuppressWarnings 'resource' to 'main()'
Configure problem severity


day09 - demo04/src/com/sunbeam/Program04.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
demo01 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Date.java
Person.java
Program01.java
demo02 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program02.java
demo03 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program03.java
demo04 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program04.java
Program04.java
12 // "sc" is not closed -- stdin in not closed -- resource leakage
13 }
14 */
15
16 public static void main(String[] args) {
17 Scanner sc = new Scanner(System.in);
18 System.out.print("Entet a number: ");
19 int num = sc.nextInt();
20 System.out.println("Square: " + num * num);
21 sc.close(); // internally close System.in i.e. stdin
22 }
23
24 }
25
if user give wrong input, program will be aborted with exception and sc.close() is not called.
in this case resource will leak.
Problems Javadoc Declaration Console
<terminated> Program04 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 12:26
Entet a number: three
Exception in thread "main" java.util.InputMismatchException
at java.base/java.util.Scanner.throwFor(Scanner.java:939)
at java.base/java.util.Scanner.next(Scanner.java:1594)
at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
at com.sunbeam.Program04.main(Program04.java:19)
Writable
Smart Insert
24 : 2 : 624
Search
12:26 PM


day09 - demo04/src/com/sunbeam/Program04.java - Spring Tool Suite 4
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
demo01 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Date.java
Person.java
Program01.java
demo02 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program02.java
demo03 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program03.java
demo04 [CJ-H-02 main]
JRE System Library [JavaSE-1
src
com.sunbeam
Program04.java
Program04.java
26 public static void main(String[] args) {
27 Scanner sc = new Scanner(System.in);
28 try {
29 System.out.print("Entet a number: ");
30 int num = sc.nextInt();
31 System.out.println("Square: " + num * num);
32 System.exit(0);
33 }
34 finally {
35 System.out.println("Closing scanner.");
36 sc.close(); // internally close System.in i.e. stdin
37 }
38 }
if application JVM exits, then finally block is not executed. Rest all cases (like return from method, break, ...) finally will execute.
finally block is always executed irrespective of exception occurrs or not.
Problems Javadoc Declaration Console
<terminated> Program04 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 12:30
Entet a number: three
Closing scanner.
Exception in thread "main" java.util.InputMismatchException
at java.base/java.util.Scanner.throwFor(Scanner.java:939)
at java.base/java.util.Scanner.next(Scanner.java:1594)
at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
at com.sunbeam.Program04.main(Program04.java:30)
Writable
Smart Insert
32 : 13 : 900
Search
12:30 PM

day09 - demo04/src/com/sunbeam/Program04.java - Spring Tool Suite 4

File Edit Source Refactor Navigate Search Project Run Window Help

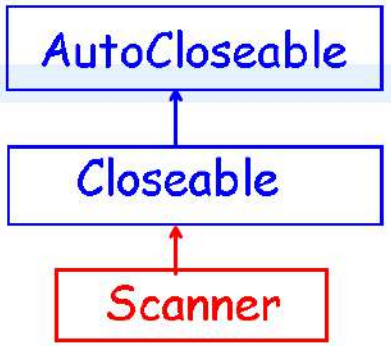
Package Explorer

- demo01 [CJ-H-02 main]
- demo02 [CJ-H-02 main]
- demo03 [CJ-H-02 main]
- demo04 [CJ-H-02 main]
  - JRE System Library [Jav
  - src
    - com.sunbeam
      - Program04.java

Program04.java

```
37              sc.close(); // internally close System.in i.e. stdin
38          }
39      }
40      */
41
42      public static void main(String[] args) {
43          // try-with-resource (since Java 7) ensure that resource is auto-closed.
44          try(Scanner sc = new Scanner(System.in)) {
45              System.out.print("Entet a number: ");
46              int num = sc.nextInt();
47              System.out.println("Square: " + num * num);
48          } // sc.close(); // called automatically
49      }
50  }
```

try-with-resource works with any class that is inherited from AutoCloseable.



Problems Javadoc Declaration Console

<terminated> Program04 [Java Application] C:\Nilesh\setup\sts-4.15.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20220515-1416\jre\bin\javaw.exe (Feb 8, 2024, 12:38:49 PM

```
Entet a number: 4
Square: 16
```

Search

12:39 PM