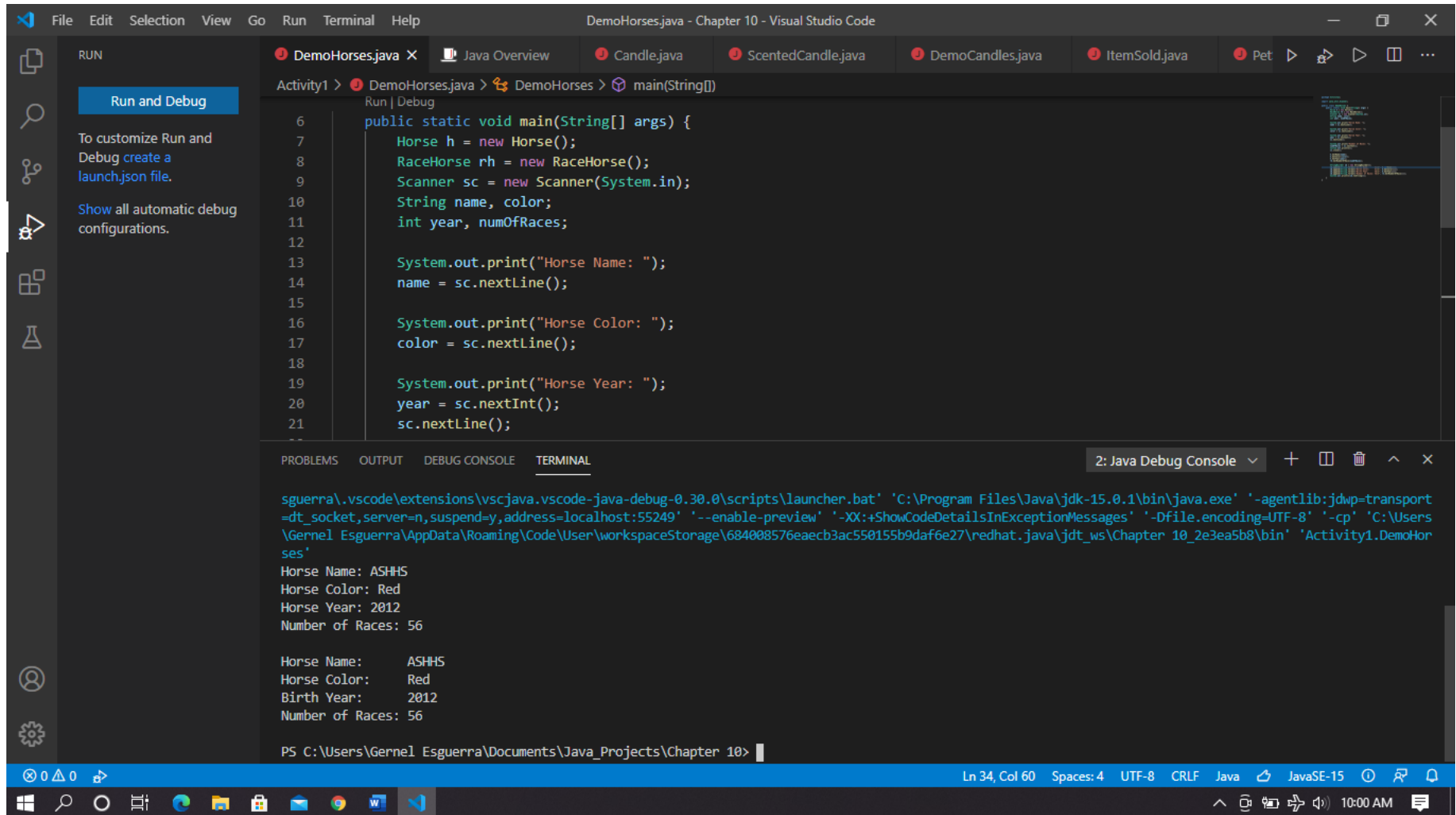


## 1. Chapter 10 – DemoHorse.java



The screenshot shows the Visual Studio Code interface with the file `DemoHorse.java` open. The code is as follows:

```
6 public static void main(String[] args) {
7     Horse h = new Horse();
8     RaceHorse rh = new RaceHorse();
9     Scanner sc = new Scanner(System.in);
10    String name, color;
11    int year, numOfRaces;
12
13    System.out.print("Horse Name: ");
14    name = sc.nextLine();
15
16    System.out.print("Horse Color: ");
17    color = sc.nextLine();
18
19    System.out.print("Horse Year: ");
20    year = sc.nextInt();
21    sc.nextLine();
22 }
```

The terminal output shows the command used to run the program and the resulting output:

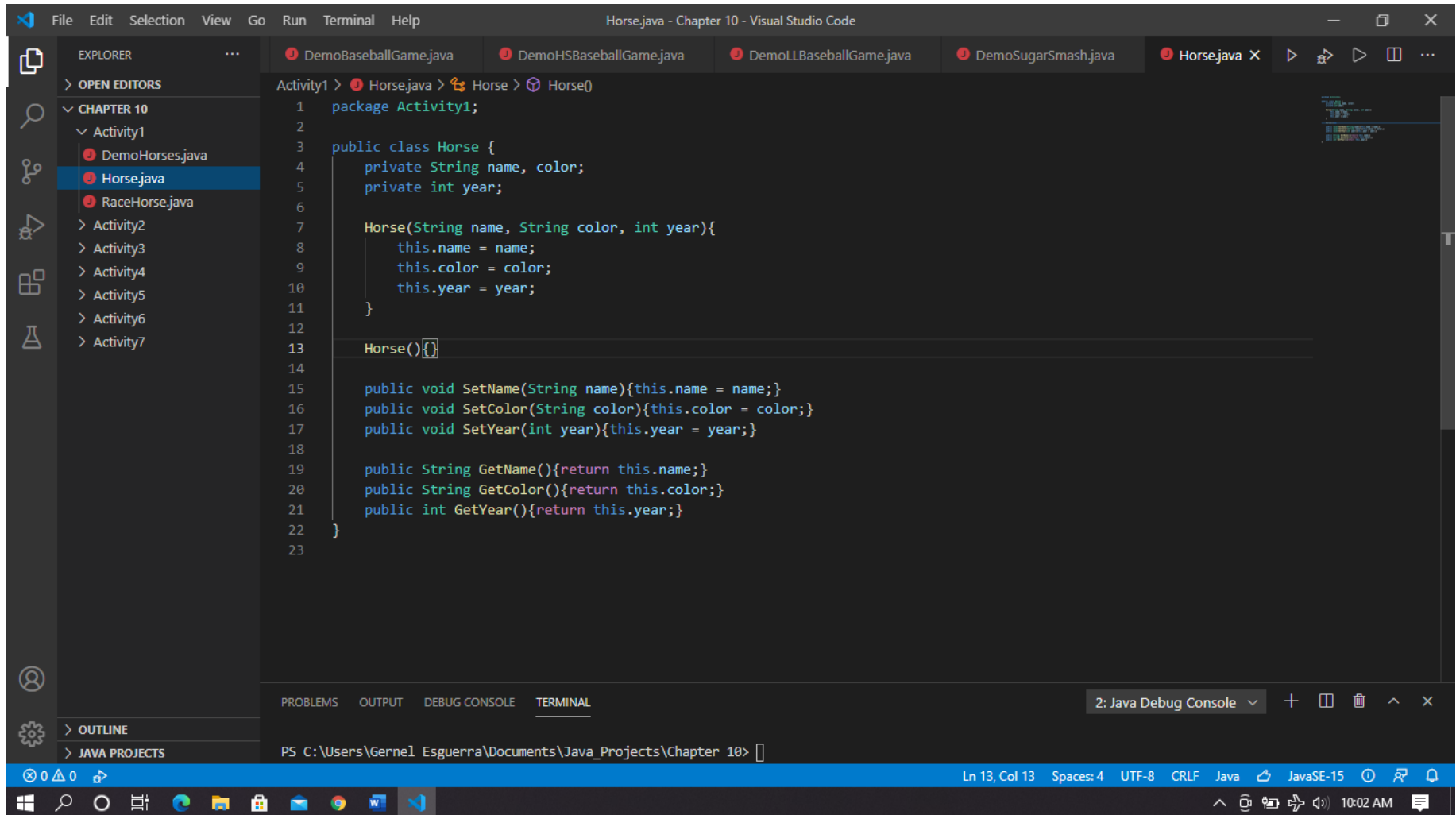
```
sguerra\.vscode\extensions\vscjava.vscode-java-debug-0.30.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk-15.0.1\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:55249' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\Gerne1 Esguerra\AppData\Roaming\Code\User\workspaceStorage\684008576eac3ac550155b9daf6e27\redhat.java\jdt_ws\Chapter_10_2e3ea5b8\bin' 'Activity1.DemoHorses'
```

```
Horse Name: ASHHS
Horse Color: Red
Horse Year: 2012
Number of Races: 56

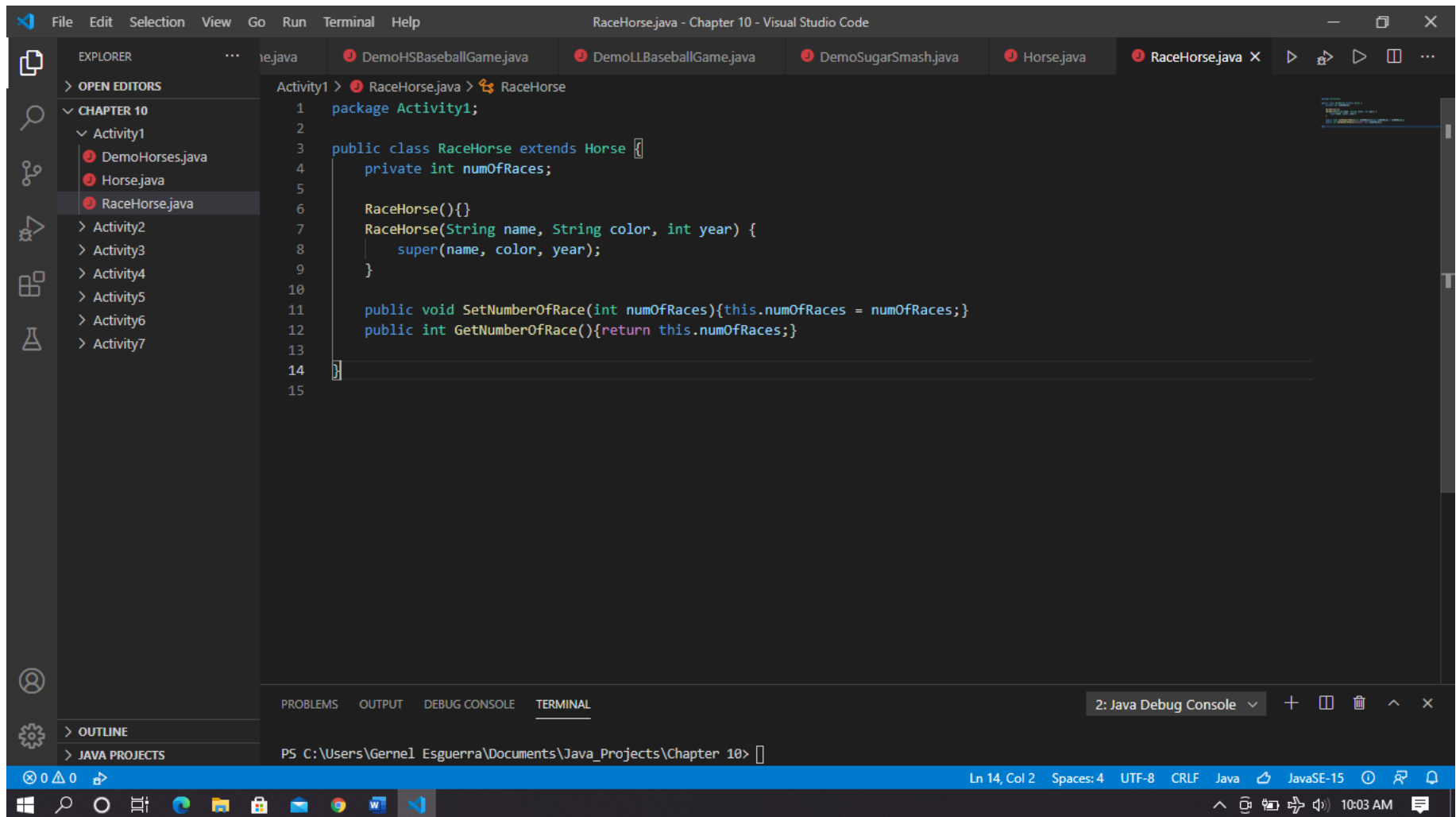
Horse Name: ASHHS
Horse Color: Red
Birth Year: 2012
Number of Races: 56
```

The terminal prompt is `PS C:\Users\Gerne1 Esguerra\Documents\Java_Projects\Chapter_10>`.

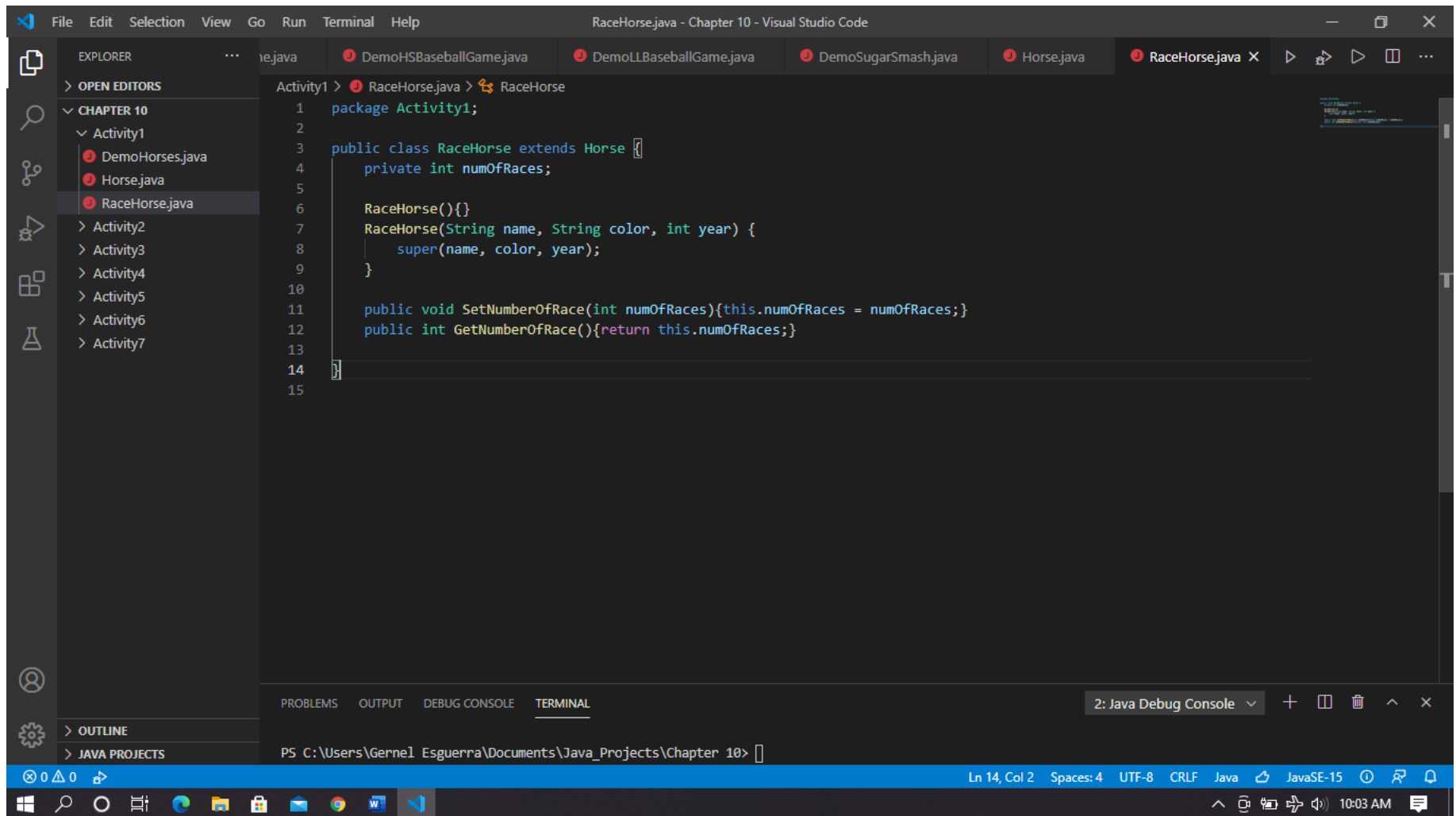
## 1. Chapter 10 – Horse.java



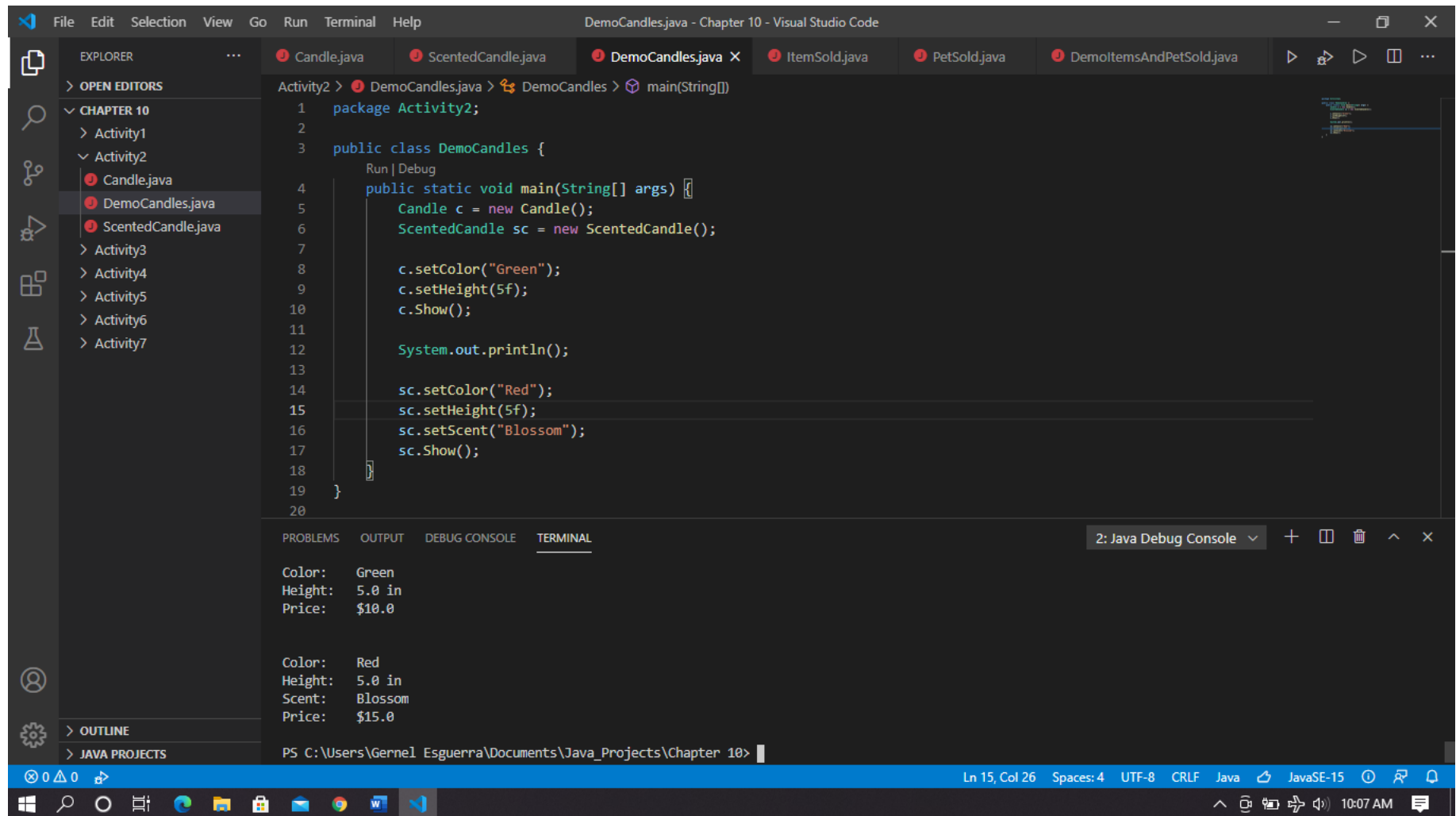
## 1. Chapter 10 – RaceHorse.java



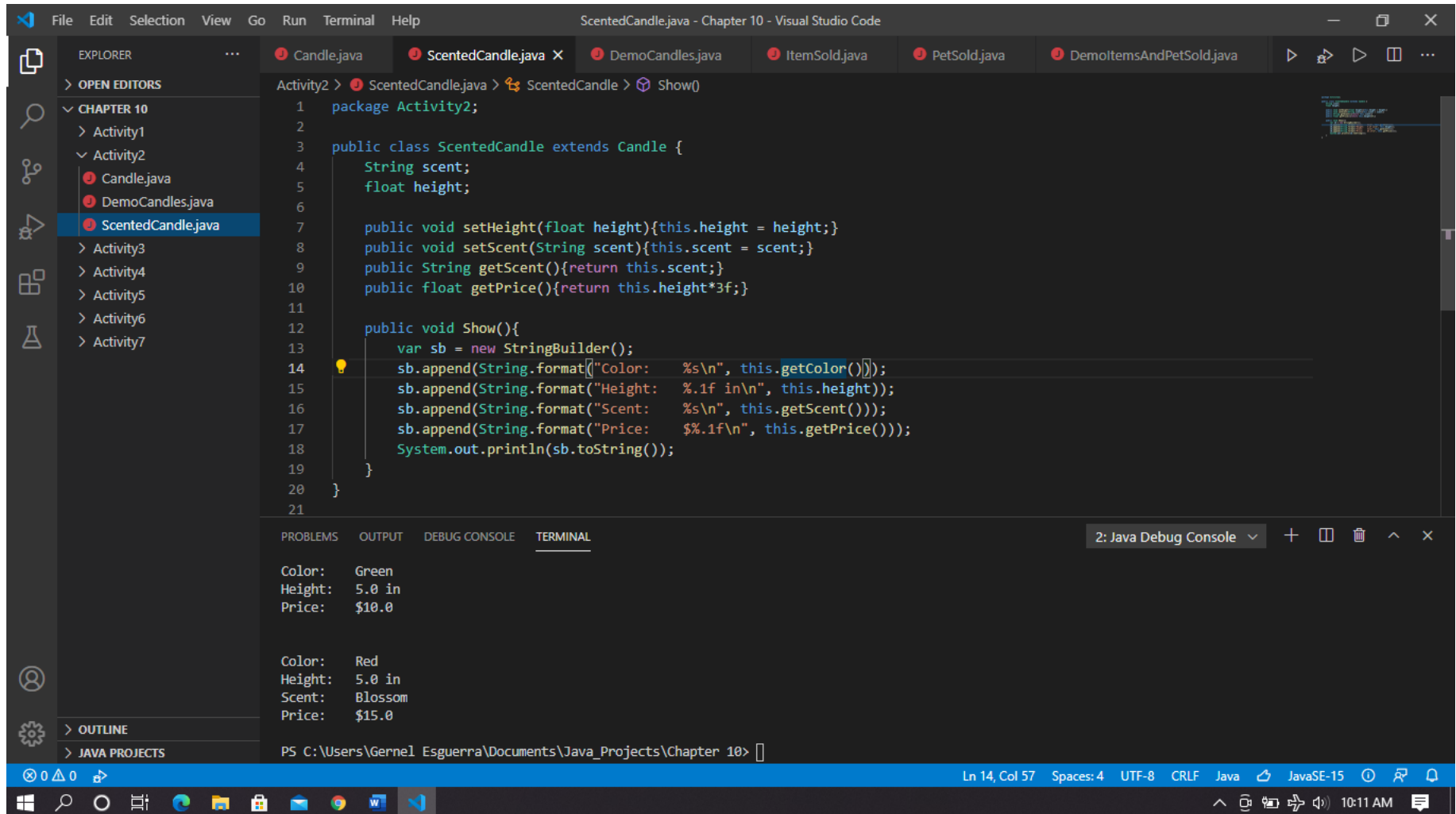
## 2. Chapter 10 – Candle.java



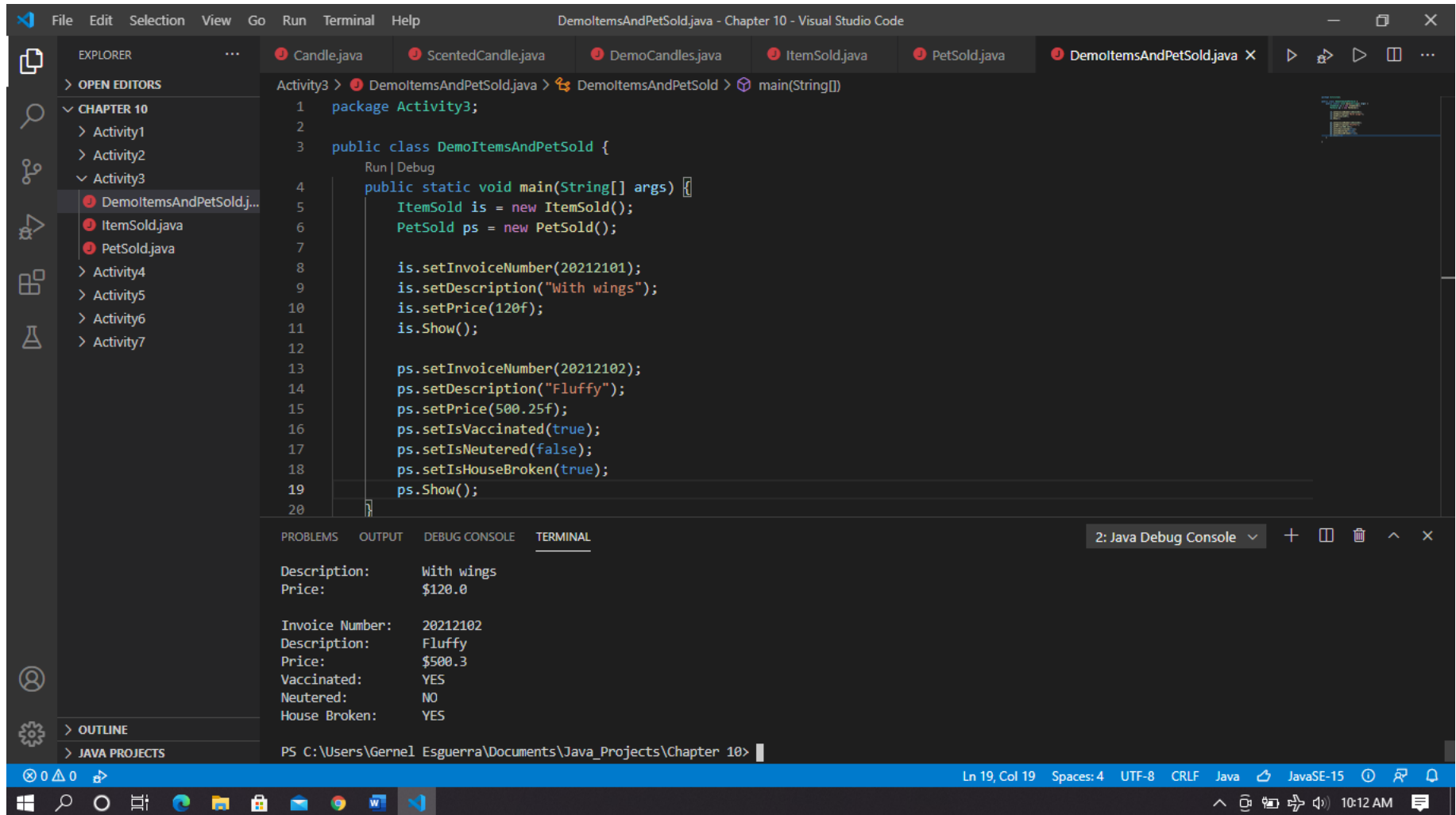
## 2. Chapter 10 – DemoCandles.java



## 2. Chapter 10 – ScentedCandles.java



### 3. Chapter 10 – DemoItemAndPetSold.java



### 3. Chapter 10 – ItemSold.java

```
File Edit Selection View Go Run Terminal Help
ItemSold.java - Chapter 10 - Visual Studio Code

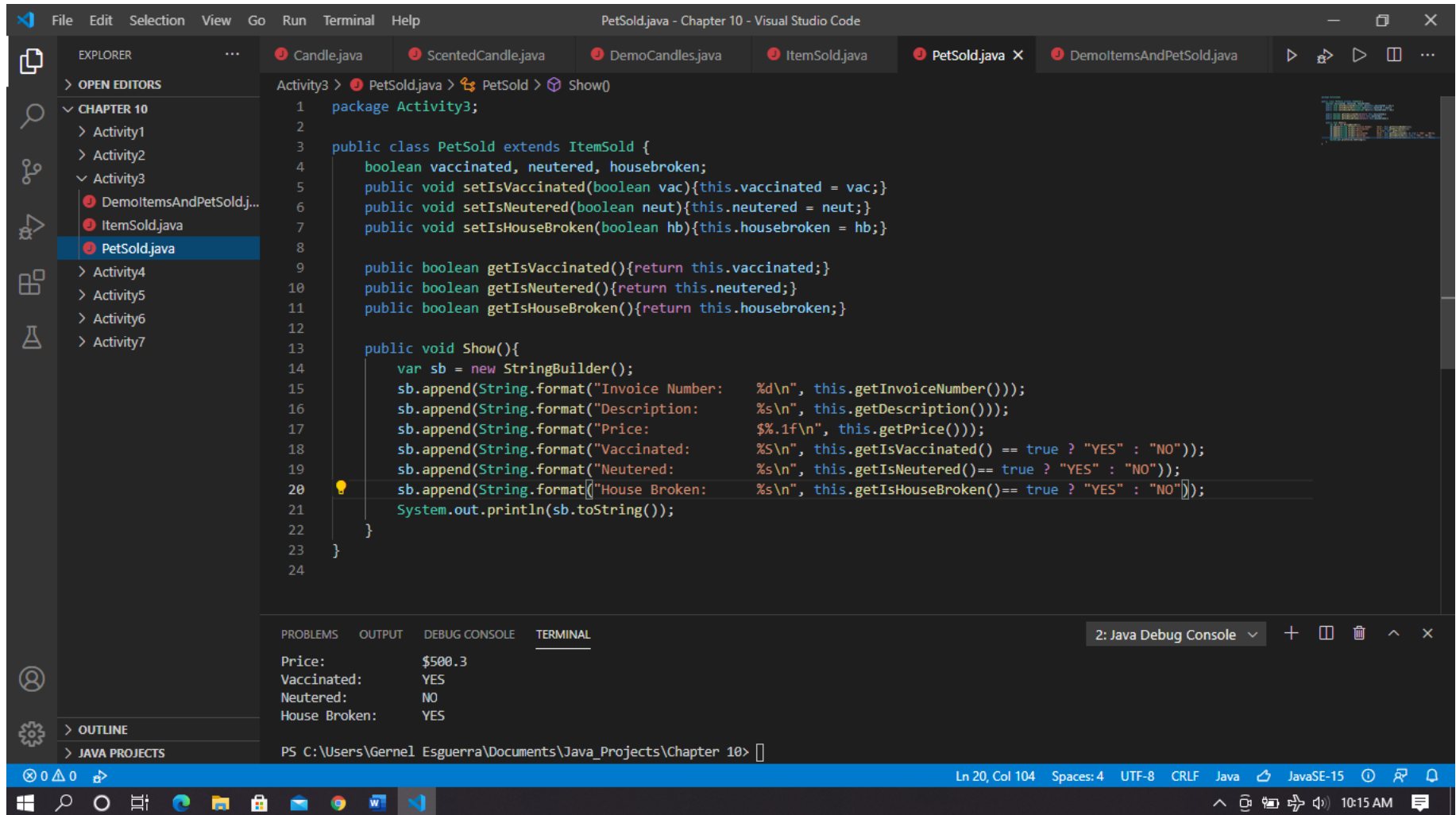
EXPLORER
> OPEN EDITORS
  > CHAPTER 10
    > Activity1
    > Activity2
    > Activity3
      > DemoItemsAndPetSold.j...
      > ItemSold.java
      > PetSold.java
    > Activity4
    > Activity5
    > Activity6
    > Activity7

Activity3 > ItemSold.java > ItemSold > Show()
1 package Activity3;
2
3 public class ItemSold {
4     int invoiceNumber;
5     String description;
6     float price;
7
8     public void setInvoiceNumber(int inum){this.invoiceNumber = inum;}
9     public void setDescription(String desc){this.description = desc;}
10    public void setPrice(float price){this.price = price;}
11
12    public int getInvoiceNumber(){return this.invoiceNumber;}
13    public String getDescription(){return this.description;}
14    public float getPrice(){return price;}
15
16    public void Show(){
17        var sb = new StringBuilder();
18        sb.append(String.format("Invoice Number:    %d\n", this.getInvoiceNumber()));
19        sb.append(String.format("Description:    %s\n", this.getDescription()));
20        sb.append(String.format("Price:        $%.1f\n", this.getPrice()));
21        System.out.println(sb.toString());
22    }
23 }
24

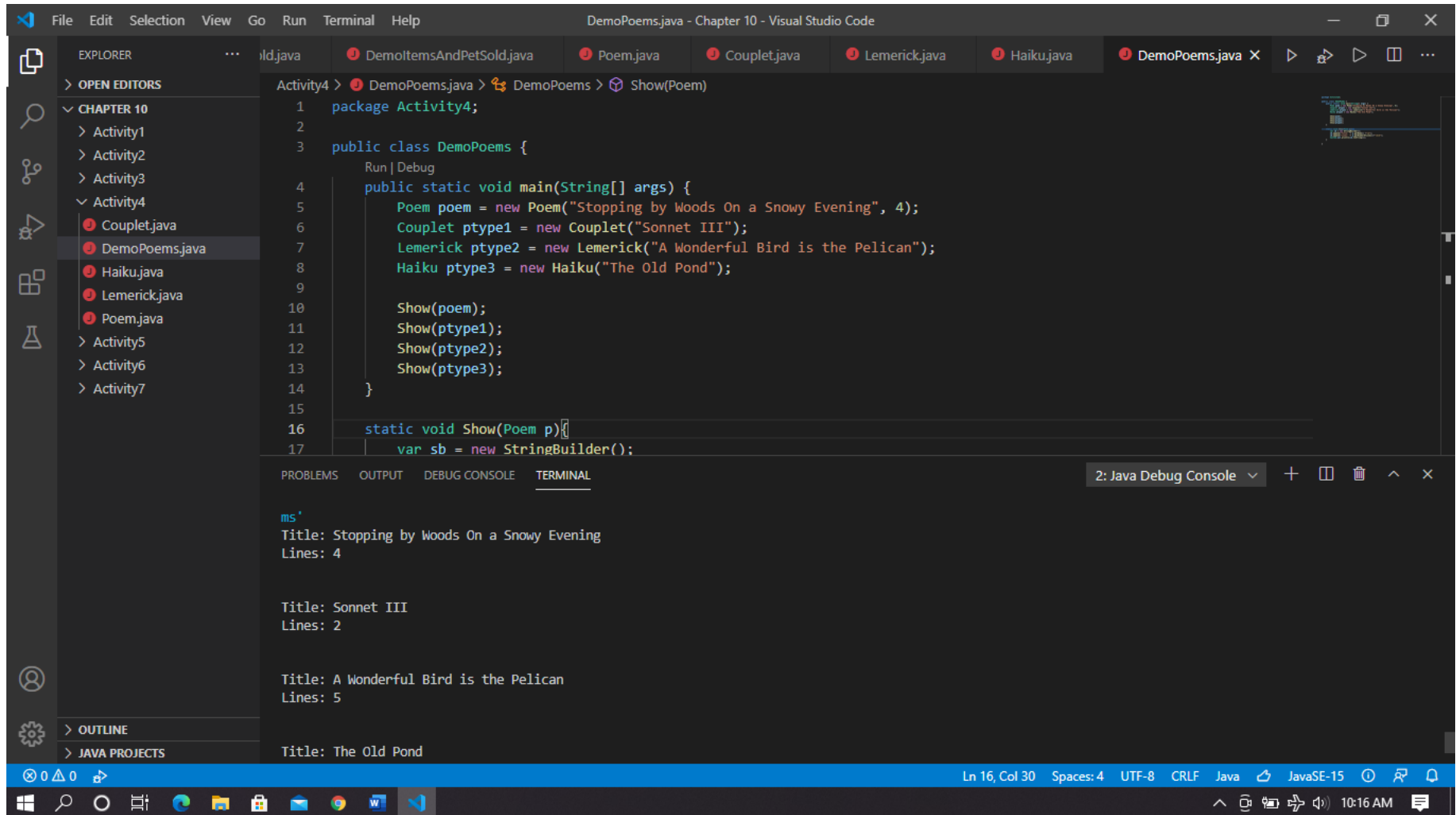
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Price: $500.3
Vaccinated: YES
Neutered: NO
House Broken: YES
PS C:\Users\Gerne1 Esguerra\Documents\Java_Projects\Chapter 10>
```



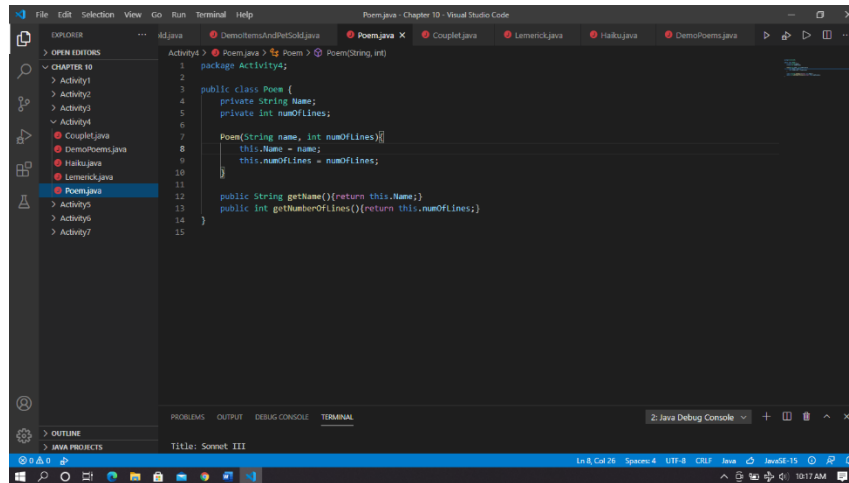
### 3. Chapter 10 - PetSold.java



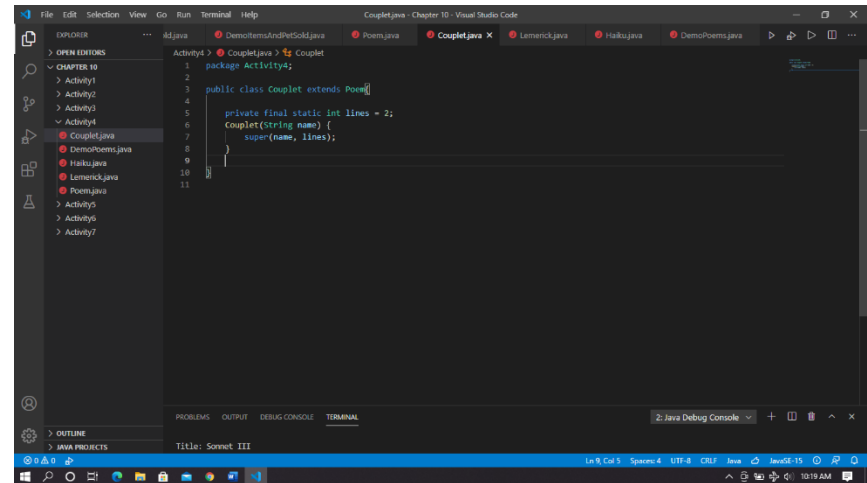
#### 4. Chapter 10 – DemoPoems.java



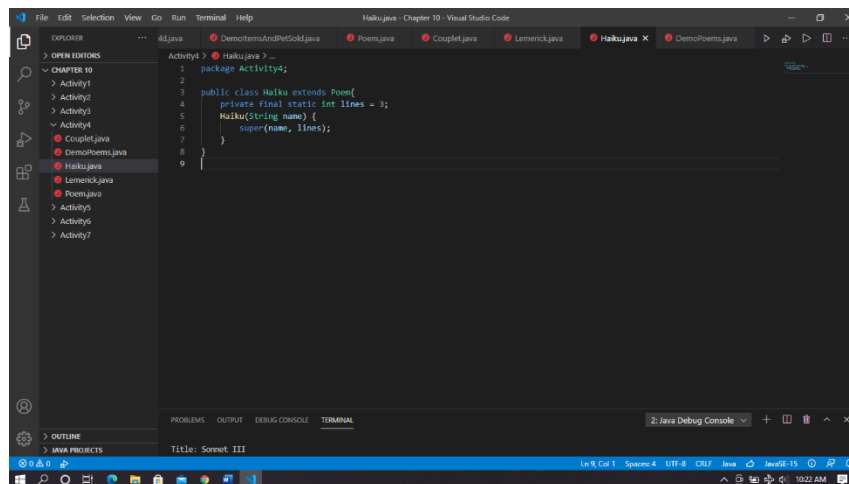
#### 4. Chapter 10 – Poems.java – Couplet.java – Haiku.java – Lemerick.java



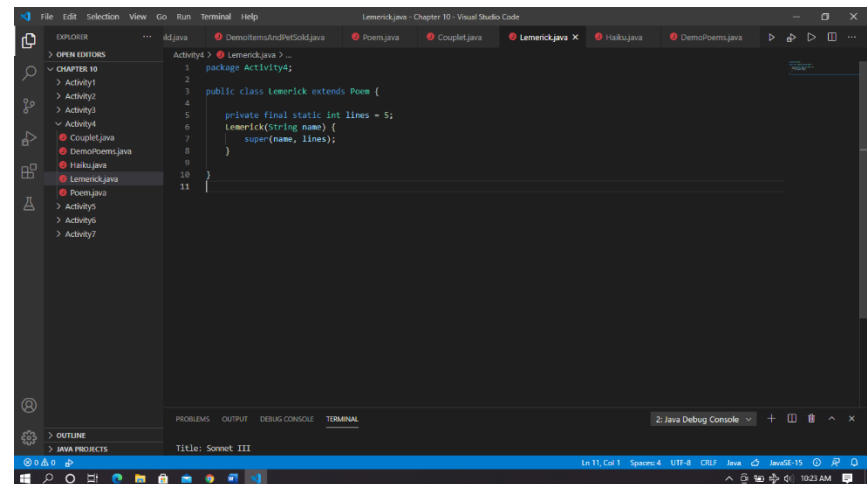
```
1 package Activity4;
2
3 public class Poem {
4     private String Name;
5     private int numOfLines;
6
7     Poem(String name, int numOfLines){
8         this.Name = name;
9         this.numOfLines = numOfLines;
10    }
11
12    public String getName(){return this.Name;}
13    public int getNumberOfLines(){return this.numOfLines;}
14
15 }
```



```
1 package Activity4;
2
3 public class Couplet extends Poem{
4
5     private final static int lines = 2;
6     Couplet(String name) {
7         super(name, lines);
8     }
9
10 }
11
```

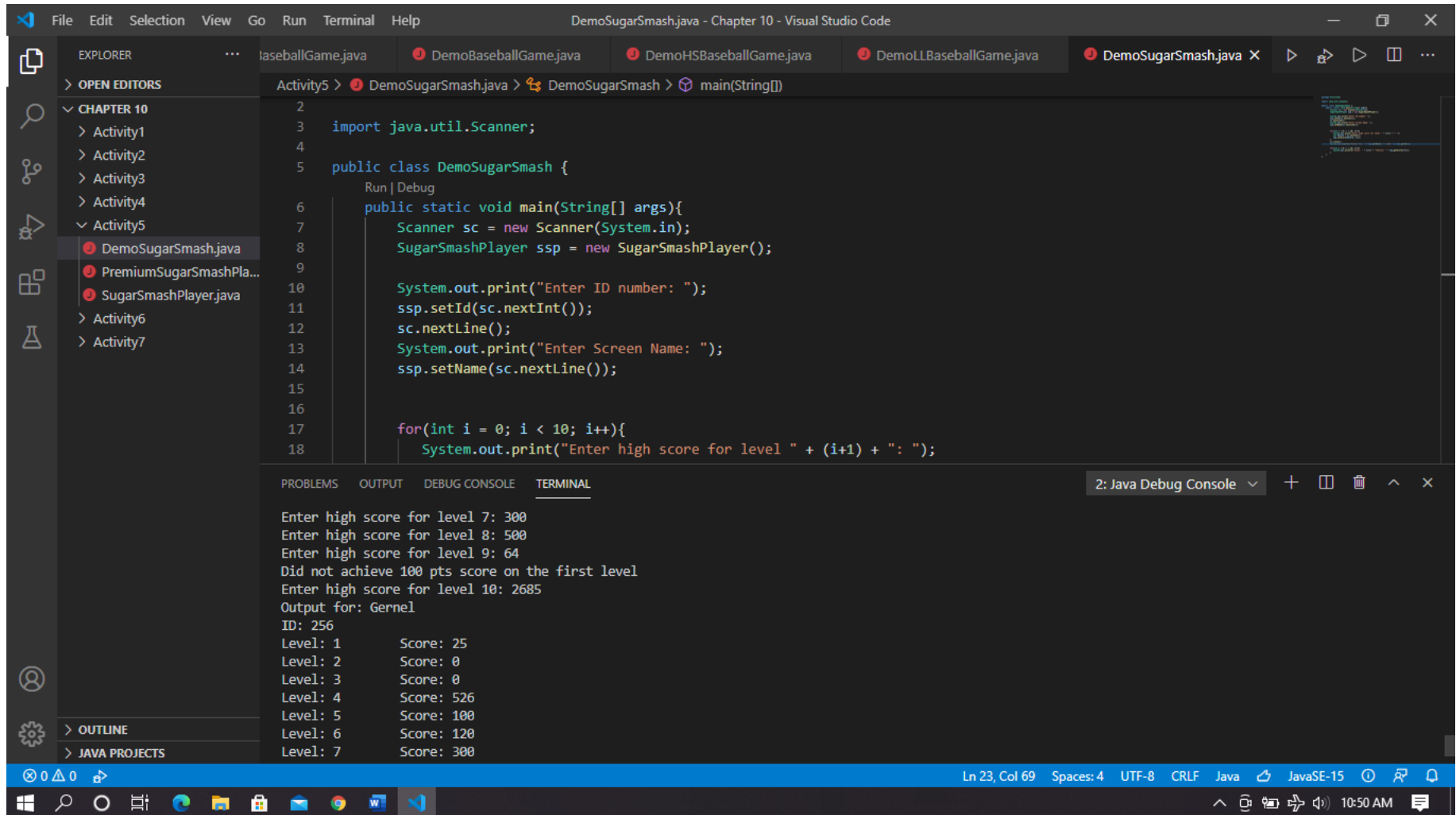


```
1 package Activity4;
2
3 public class Haiku extends Poem{
4     private final static int lines = 3;
5     Haiku(String name) {
6         super(name, lines);
7     }
8
9 }
```



```
1 package Activity4;
2
3 public class Lemerick extends Poem {
4
5     private final static int lines = 5;
6     Lemerick(String name) {
7         super(name, lines);
8     }
9
10 }
11
```

## 5. Chapter 10 – DemoSugarSmash.java



The screenshot displays the Visual Studio Code interface with the file `DemoSugarSmash.java` open. The Explorer sidebar on the left shows the project structure, including `CHAPTER 10` and its sub-activities. The main editor area shows the Java code for `DemoSugarSmash`, which uses a `Scanner` to collect user input for ID and name, and a `for` loop to prompt for high scores across 10 levels.

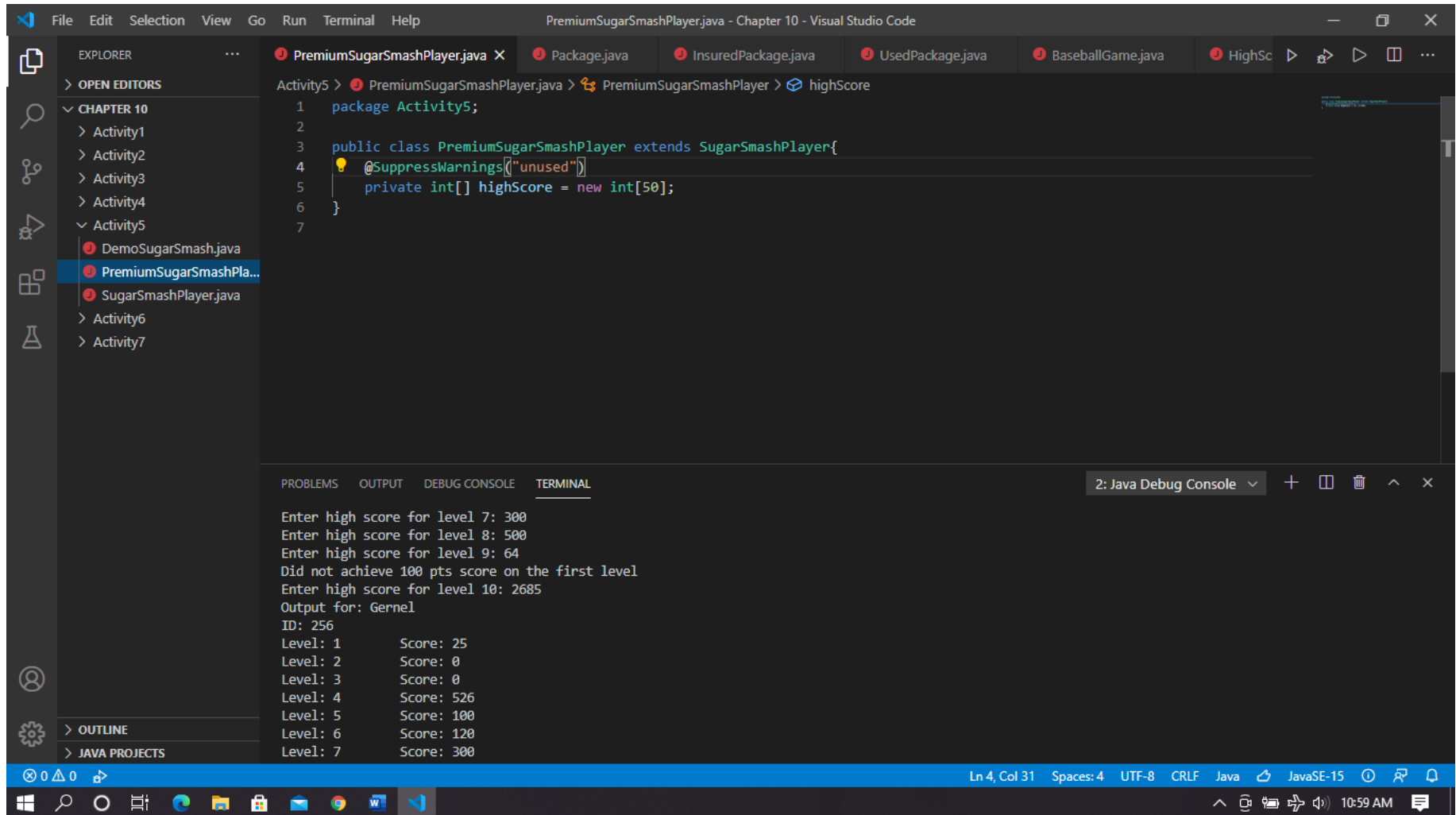
```
2
3 import java.util.Scanner;
4
5 public class DemoSugarSmash {
6     Run | Debug
7     public static void main(String[] args){
8         Scanner sc = new Scanner(System.in);
9         SugarSmashPlayer ssp = new SugarSmashPlayer();
10
11         System.out.print("Enter ID number: ");
12         ssp.setId(sc.nextInt());
13         sc.nextLine();
14         System.out.print("Enter Screen Name: ");
15         ssp.setName(sc.nextLine());
16
17         for(int i = 0; i < 10; i++){
18             System.out.print("Enter high score for level " + (i+1) + ": ");
```

The bottom panel shows the `TERMINAL` output, which includes the program's execution flow, user input for ID (256) and name (Gernel), and a list of scores for levels 1 through 7.

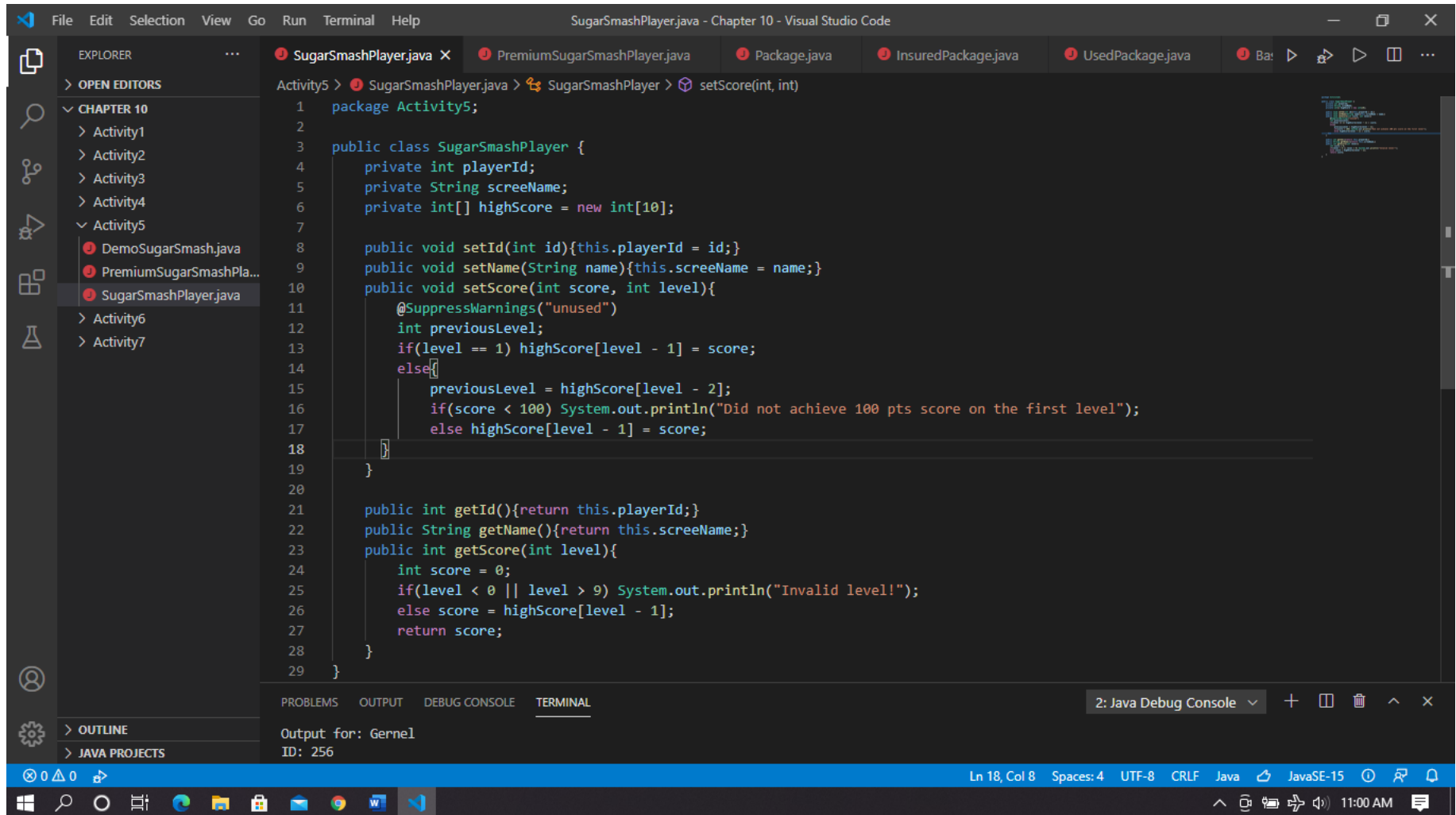
```
Enter high score for level 7: 300
Enter high score for level 8: 500
Enter high score for level 9: 64
Did not achieve 100 pts score on the first level
Enter high score for level 10: 2685
Output for: Gernel
ID: 256
Level: 1      Score: 25
Level: 2      Score: 0
Level: 3      Score: 0
Level: 4      Score: 526
Level: 5      Score: 100
Level: 6      Score: 120
Level: 7      Score: 300
```

The status bar at the bottom indicates the current position is `Ln 23, Col 69` with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the `Java` language mode.

## 5. Chapter 10 – PremiumSugarSmashPlayer.java



## 5. Chapter 10 – SugarSmashPlayer.java



```
File Edit Selection View Go Run Terminal Help
SugarSmashPlayer.java - Chapter 10 - Visual Studio Code

EXPLORER
> OPEN EDITORS
  > CHAPTER 10
    > Activity1
    > Activity2
    > Activity3
    > Activity4
    > Activity5
      DemoSugarSmash.java
      PremiumSugarSmashPla...
      SugarSmashPlayer.java
    > Activity6
    > Activity7

> OUTLINE
> JAVA PROJECTS

Activity5 > SugarSmashPlayer.java > SugarSmashPlayer > setScore(int, int)
1 package Activity5;
2
3 public class SugarSmashPlayer {
4     private int playerId;
5     private String screenName;
6     private int[] highScore = new int[10];
7
8     public void setId(int id){this.playerId = id;}
9     public void setName(String name){this.screenName = name;}
10    public void setScore(int score, int level){
11        @SuppressWarnings("unused")
12        int previousLevel;
13        if(level == 1) highScore[level - 1] = score;
14        else{
15            previousLevel = highScore[level - 2];
16            if(score < 100) System.out.println("Did not achieve 100 pts score on the first level");
17            else highScore[level - 1] = score;
18        }
19    }
20
21    public int getId(){return this.playerId;}
22    public String getName(){return this.screenName;}
23    public int getScore(int level){
24        int score = 0;
25        if(level < 0 || level > 9) System.out.println("Invalid level!");
26        else score = highScore[level - 1];
27        return score;
28    }
29 }
```

2: Java Debug Console

Output for: Gernel  
ID: 256

Ln 18, Col 8 Spaces: 4 UTF-8 CRLF Java JavaSE-15 11:00 AM

## 6. Chapter 10 – BaseballGame.java

```
File Edit Selection View Go Run Terminal Help
BaseballGame.java - Chapter 10 - Visual Studio Code

EXPLORER
> OPEN EDITORS
  Activity6 > BaseballGame.java > BaseballGame > computeScore()
  CHAPTER 10
    > Activity1
    > Activity2
    > Activity3
    > Activity4
    > Activity5
    > Activity6
      BaseballGame.java
      DemoBaseballGame.java
      DemoHSBaseballGame.j...
      DemoLLBaseballGame.j...
      HighSchoolBaseballGa...
      LittleLeagueBaseballGa...
    > Activity7
      InsuredPackage.java
      Package.java
      UsedPackage.java

> OUTLINE
> JAVA PROJECTS

4 private String[] teams = new String[2];
5 protected int[][] scores;
6
7 public BaseballGame(){
8     scores = new int[2][9];
9     for(int i = 0; i < 2; i++)
10         for(int j = 0; j < 9; j++)
11             scores[i][j] = 999;
12 }
13
14 public void setTeams(int team, String name){
15     teams[team] = name;
16 }
17
18 public void setScores(int position, int teamNumber, int score){
19     if(position == 0 || (position < sco int score - Activity6.BaseballGame.setScores(int, int, int) 999){
20         scores[teamNumber][position] = score;
21         if(scores[0][scores[0].length-1] != 999 && scores[1][scores[1].length-1] != 999)
22             computeScore();
23     }
24 }
25
26 private void computeScore() {
27     int a = 0, b = 0;
28     for(int score: scores[0])
29         a += score;
30
31     for(int score: scores[1])
32         b += score;
33 }
```

2: Java Debug Console

Output for: Gernel  
ID: 256

Ln 36, Col 36 Spaces: 4 UTF-8 CRLF Java JavaSE-15 11:04 AM

## 6. Chapter 10 – DemoBaseballGame.java

The screenshot displays the Visual Studio Code interface with the file `DemoBaseballGame.java` open. The Explorer sidebar on the left shows a project structure with `CHAPTER 10` expanded, listing `Activity1` through `Activity7`. Under `Activity6`, several Java files are listed, including `DemoBaseballGame.java`, which is the active file. The main editor shows the code for `DemoBaseballGame.java`, which includes package declarations, imports, and a `main` method that uses a `Scanner` to read input and a `BaseballGame` object to manage the game state. The `main` method includes a loop for entering scores for two teams over nine innings. The bottom panel shows the `TERMINAL` tab with the output of the program, showing the user entering scores for the 2nd team in each inning, and the final result: `Team 2 Wins!`. The status bar at the bottom indicates the current line and column (Ln 16, Col 23) and the encoding (UTF-8).

```
File Edit Selection View Go Run Terminal Help
DemoBaseballGame.java - Chapter 10 - Visual Studio Code

EXPLORER
> OPEN EDITORS
  DemoBaseballGame.java
  DemoBaseballGame.java
  DemoHSBaseballGame.j...
  DemoLLBaseballGame.j...
  HighSchoolBaseballGa...
  LittleLeaugeBaseballGa...
  Activity7
  InsuredPackage.java
  Package.java
  UsedPackage.java

> CHAPTER 10
  > Activity1
  > Activity2
  > Activity3
  > Activity4
  > Activity5
  > Activity6
    BaseballGame.java
    DemoBaseballGame.java
    DemoHSBaseballGame.j...
    DemoLLBaseballGame.j...
    HighSchoolBaseballGa...
    LittleLeaugeBaseballGa...
  > Activity7
    InsuredPackage.java
    Package.java
    UsedPackage.java

> OUTLINE
> JAVA PROJECTS

Activity6 > DemoBaseballGame.java > DemoBaseballGame > main(String[])
1 package Activity6;
2
3 import java.util.Scanner;
4
5 public class DemoBaseballGame {
6     Run | Debug
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         int count;
10
11         String[] team = {"1st", "2nd"};
12
13         BaseballGame b = new BaseballGame();
14         for(int i = 0; i < 2; i++){
15             System.out.print("\nInput " + team[i] + " Team >> ");
16             b.setTeams(i, sc.nextLine());
17             count = 0;
18             while(count < 9){
19                 System.out.print(String.format("Enter the score of %s team in inning %d >> ", team[i], count+1));
20                 b.setScores(count, i, sc.nextInt());
21                 sc.nextLine();
22             }
23         }
24     }
25 }
```

2: Java Debug Console

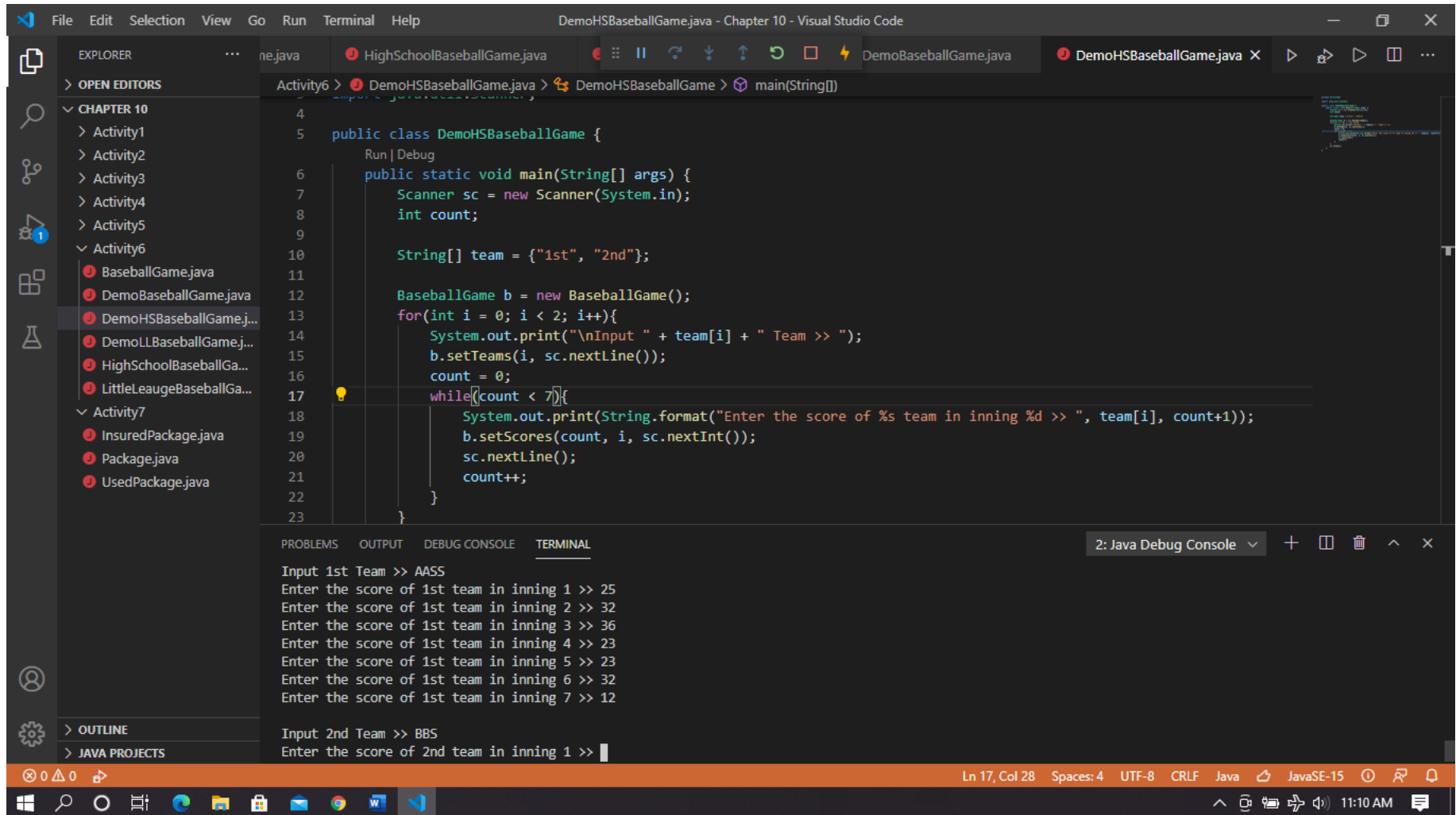
```
Enter the score of 2nd team in inning 2 >> 65
Enter the score of 2nd team in inning 3 >> 123
Enter the score of 2nd team in inning 4 >> 23
Enter the score of 2nd team in inning 5 >> 14
Enter the score of 2nd team in inning 6 >> 35
Enter the score of 2nd team in inning 7 >> 1
Enter the score of 2nd team in inning 8 >> 1
Enter the score of 2nd team in inning 9 >> 1

Team 2 Wins!
PS C:\Users\Gerne1 Esguerra\Documents\Java_Projects\Chapter 10>
```

Ln 16, Col 23 Spaces: 4 UTF-8 CRLF Java JavaSE-15 11:07 AM



## 6. Chapter 10 – DemoHSBaseballGame.java



The screenshot displays the Visual Studio Code interface with the file `DemoHSBaseballGame.java` open. The Explorer sidebar on the left shows a project structure with `CHAPTER 10` containing `Activity1` through `Activity7`. Under `Activity6`, several Java files are listed, including `DemoHSBaseballGame.java`. The main editor window shows the code for `DemoHSBaseballGame`, which includes a `main` method that uses a `Scanner` to read team names and scores for two teams over seven innings. The `main` method is as follows:

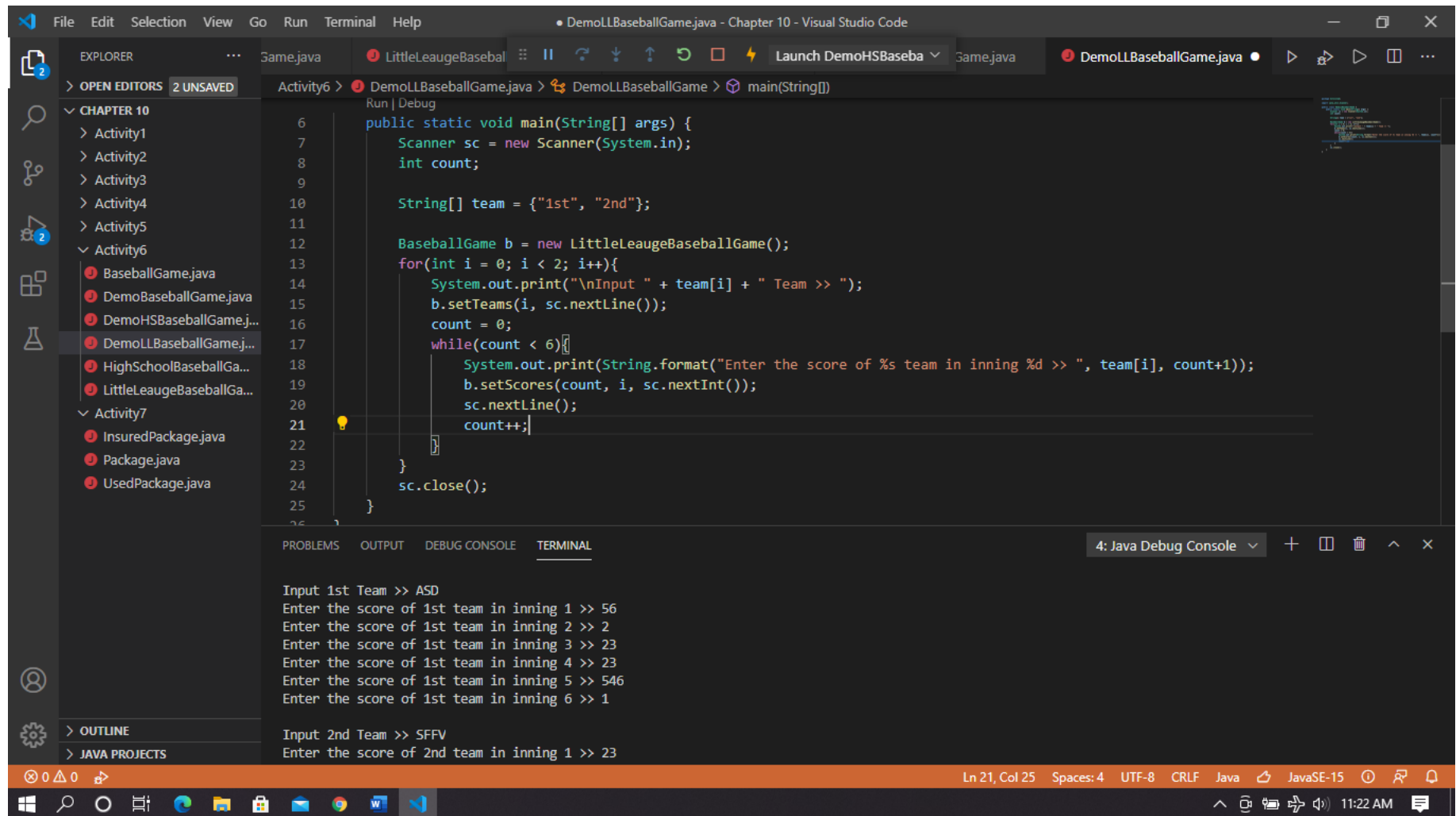
```
4  
5 public class DemoHSBaseballGame {  
6     public static void main(String[] args) {  
7         Scanner sc = new Scanner(System.in);  
8         int count;  
9  
10        String[] team = {"1st", "2nd"};  
11  
12        BaseballGame b = new BaseballGame();  
13        for(int i = 0; i < 2; i++){  
14            System.out.print("\nInput " + team[i] + " Team >> ");  
15            b.setTeams(i, sc.nextLine());  
16            count = 0;  
17            while(count < 7){  
18                System.out.print(String.format("Enter the score of %s team in inning %d >> ", team[i], count+1));  
19                b.setScores(count, i, sc.nextInt());  
20                sc.nextLine();  
21                count++;  
22            }  
23        }  
24    }  
25 }
```

The bottom of the screen shows the `TERMINAL` tab with the following output:

```
Input 1st Team >> AASS  
Enter the score of 1st team in inning 1 >> 25  
Enter the score of 1st team in inning 2 >> 32  
Enter the score of 1st team in inning 3 >> 36  
Enter the score of 1st team in inning 4 >> 23  
Enter the score of 1st team in inning 5 >> 23  
Enter the score of 1st team in inning 6 >> 32  
Enter the score of 1st team in inning 7 >> 12  
  
Input 2nd Team >> BBS  
Enter the score of 2nd team in inning 1 >> 
```

The status bar at the bottom indicates the current position is `Ln 17, Col 28` with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the `Java` language mode.

## 6. Chapter 10 – DemoLLBaseballGame.java



```
File Edit Selection View Go Run Terminal Help
• DemoLLBaseballGame.java - Chapter 10 - Visual Studio Code

EXPLORER
2 OPEN EDITORS 2 UNSAVED
CHAPTER 10
  > Activity1
  > Activity2
  > Activity3
  > Activity4
  > Activity5
  > Activity6
    1 BaseballGame.java
    2 DemoBaseballGame.java
    3 DemoHSBaseballGame.j...
    4 DemoLLBaseballGame.j...
    5 HighSchoolBaseballGa...
    6 LittleLeaugeBaseballGa...
  > Activity7
    7 InsuredPackage.java
    8 Package.java
    9 UsedPackage.java
  > OUTLINE
  > JAVA PROJECTS

Activity6 > DemoLLBaseballGame.java > DemoLLBaseballGame > main(String[])
Run | Debug

6 public static void main(String[] args) {
7     Scanner sc = new Scanner(System.in);
8     int count;
9
10    String[] team = {"1st", "2nd"};
11
12    BaseballGame b = new LittleLeaugeBaseballGame();
13    for(int i = 0; i < 2; i++){
14        System.out.print("\nInput " + team[i] + " Team >> ");
15        b.setTeams(i, sc.nextLine());
16        count = 0;
17        while(count < 6){
18            System.out.print(String.format("Enter the score of %s team in inning %d >> ", team[i], count+1));
19            b.setScores(count, i, sc.nextInt());
20            sc.nextLine();
21            count++;
22        }
23    }
24    sc.close();
25 }
```

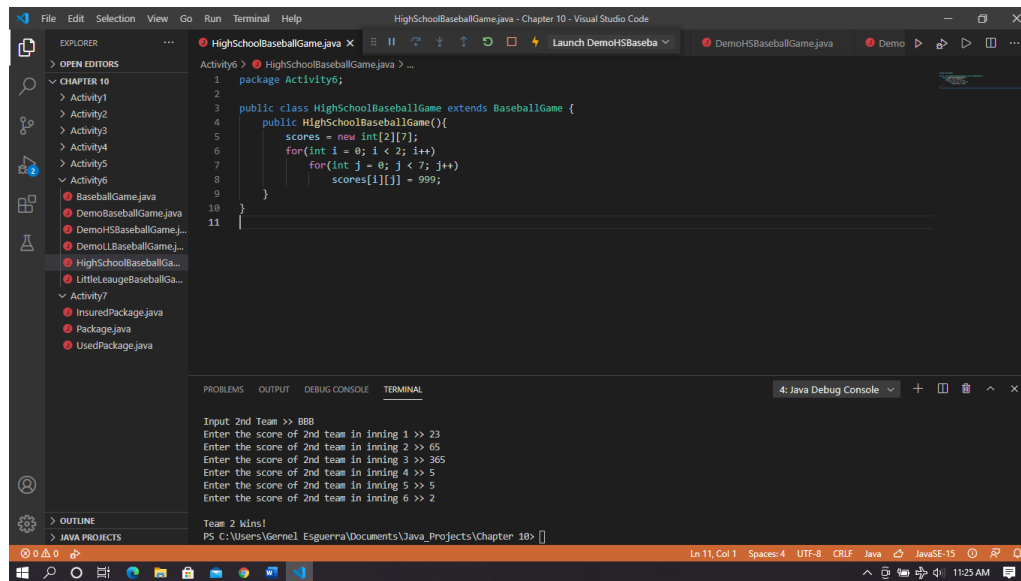
4: Java Debug Console

```
Input 1st Team >> ASD
Enter the score of 1st team in inning 1 >> 56
Enter the score of 1st team in inning 2 >> 2
Enter the score of 1st team in inning 3 >> 23
Enter the score of 1st team in inning 4 >> 23
Enter the score of 1st team in inning 5 >> 546
Enter the score of 1st team in inning 6 >> 1

Input 2nd Team >> SFFV
Enter the score of 2nd team in inning 1 >> 23
```

Ln 21, Col 25 Spaces: 4 UTF-8 CRLF Java JavaSE-15 11:22 AM

## 6. Chapter 10 – HighSchoolBaseBallGame.java – LittleLeagueBaseballGame.java

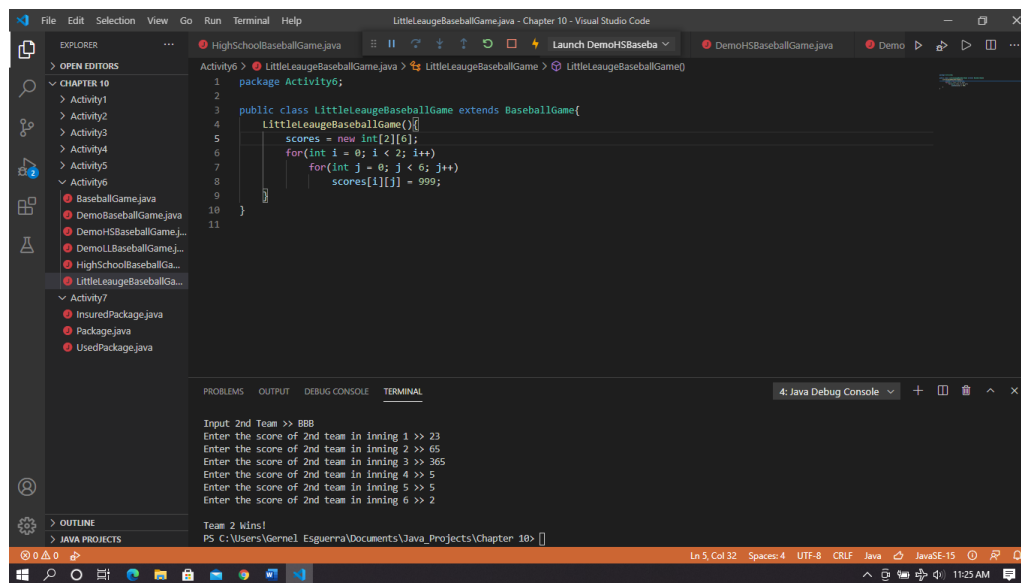


This screenshot shows the Visual Studio Code editor with the file `HighSchoolBaseBallGame.java` open. The Explorer sidebar on the left shows a project structure for Chapter 10, including `Activity1` through `Activity7`, and several other Java files like `BaseballGame.java`, `DemoBaseBallGame.java`, and `UsedPackage.java`. The main editor area displays the code for `HighSchoolBaseBallGame`, which extends `BaseballGame`. The code includes a `public HighSchoolBaseBallGame()` constructor that initializes a `scores` array and contains nested loops for inputting scores for 2 teams over 7 innings. The `main` method prompts the user to enter scores for the 2nd team in each of the 7 innings. The output window at the bottom shows the execution results, including the input scores and the final result: "Team 2 Wins!".

```
1 package Activity6;
2
3 public class HighSchoolBaseBallGame extends BaseballGame {
4     public HighSchoolBaseBallGame(){
5         scores = new int[2][7];
6         for(int i = 0; i < 2; i++)
7             for(int j = 0; j < 7; j++)
8                 scores[i][j] = 999;
9     }
10 }
11
```

Input 2nd Team >> BBB  
Enter the score of 2nd team in Inning 1 >> 23  
Enter the score of 2nd team in Inning 2 >> 65  
Enter the score of 2nd team in Inning 3 >> 365  
Enter the score of 2nd team in Inning 4 >> 5  
Enter the score of 2nd team in Inning 5 >> 5  
Enter the score of 2nd team in Inning 6 >> 2

Team 2 Wins!  
PS C:\Users\Germe1\Esguerra\Documents\Java\_Projects\Chapter 10>



This screenshot shows the Visual Studio Code editor with the file `LittleLeagueBaseballGame.java` open. The Explorer sidebar on the left shows the same project structure as the previous screenshot. The main editor area displays the code for `LittleLeagueBaseBallGame`, which extends `BaseballGame`. The code includes a `public LittleLeagueBaseBallGame()` constructor that initializes a `scores` array and contains nested loops for inputting scores for 2 teams over 6 innings. The `main` method prompts the user to enter scores for the 2nd team in each of the 6 innings. The output window at the bottom shows the execution results, including the input scores and the final result: "Team 2 Wins!".

```
1 package Activity6;
2
3 public class LittleLeagueBaseBallGame extends BaseballGame{
4     LittleLeagueBaseBallGame(){
5         scores = new int[2][6];
6         for(int i = 0; i < 2; i++)
7             for(int j = 0; j < 6; j++)
8                 scores[i][j] = 999;
9     }
10 }
11
```

Input 2nd Team >> BBB  
Enter the score of 2nd team in Inning 1 >> 23  
Enter the score of 2nd team in Inning 2 >> 65  
Enter the score of 2nd team in Inning 3 >> 365  
Enter the score of 2nd team in Inning 4 >> 5  
Enter the score of 2nd team in Inning 5 >> 5  
Enter the score of 2nd team in Inning 6 >> 2

Team 2 Wins!  
PS C:\Users\Germe1\Esguerra\Documents\Java\_Projects\Chapter 10>