

☐

I'm not robot

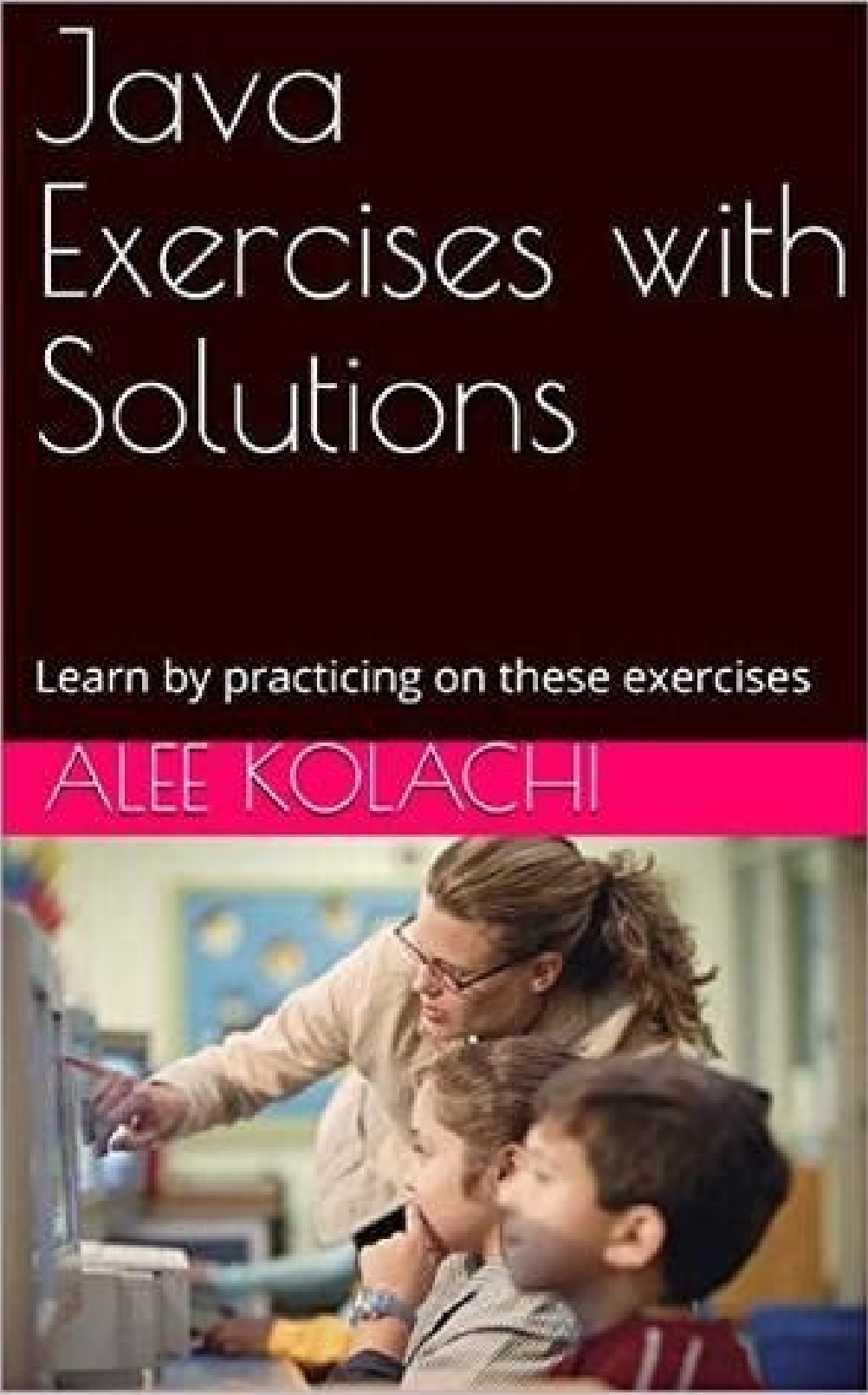
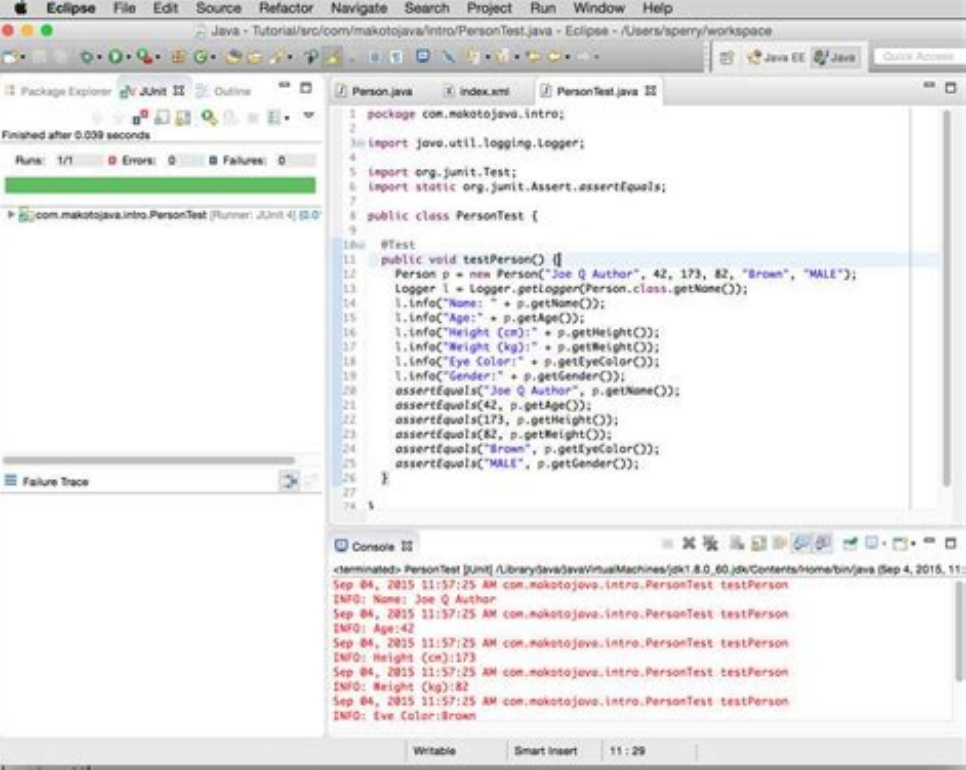

reCAPTCHA

Continue

Java exercises with solutions pdf

Java nested loop exercises with solutions pdf. Java oop inheritance exercises with solutions pdf. Java array exercises with solutions pdf. Java lab exercises with solutions pdf. Java programming exercises with solutions pdf free download. Java exercises for practice with solutions pdf. Java programming exercises with solutions pdf. Javascript practice exercises with solutions pdf. Java oop exercises with solutions pdf. Java workbook exercises for beginners with solutions pdf. Object oriented programming with java lab exercises and solutions pdf. Java while loop exercises with solutions pdf.

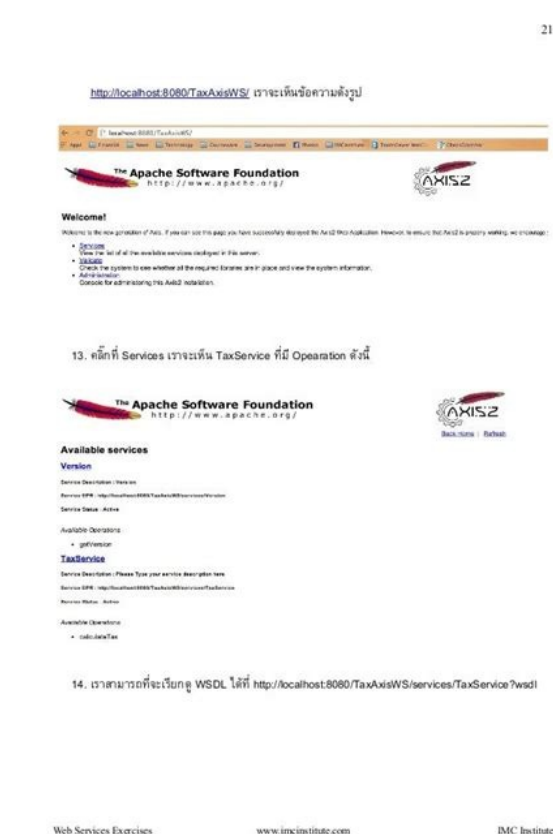
Last update on May 30 2023 15:11:55 (UTC/GMT +8 hours) Java is the foundation for virtually every type of networked application and is the global standard for developing and delivering embedded and mobile applications, games, Web-based content, and enterprise software. With more than 9 million developers worldwide, Java enables you to efficiently develop, deploy and use exciting applications and services. The best way we learn anything is by practice and exercise questions.

[illegible]

```

// Check if the above actions should be written to the console. import java.util.Random; class Main { public static void main(String[] args) { int min = 1; int max = 100; //Generating a random number from range r = new Random(); int randomNumber = min + r.nextInt(max); System.out.println("Generated random number is: "+randomNumber); //Checking whether the number is odd or even if(randomNumber%2==0){ System.out.println("The generated random number is even."); } else { System.out.println("The generated random number is odd."); } } } //Output:Generated random number is: 25 The generated random number is odd. Choose an odd number between 50 and 100 and save it as an int variable telling us how many Lego bricks we have (e.g. amountOfBricks ), then select an even number between 5 and 10 stating how many Lego blocks fit in one container (e.g., containerCapacity ) and save it as an int variable as well. Write a program that will calculate how many full containers we have, how many containers, in general, are full and not full, and how many blocks are in the container that is not completely full (use the modulo operator for this). import java.util.Scanner; class Main { public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.println("How many Lego bricks do we have? Choose an odd number between 50 and 100: "); int amountOfBricks = Integer.parseInt(sc.nextLine()); System.out.println("How many Lego blocks fit in one container? Choose an even number between 5 and 10: "); int containerCapacity = Integer.parseInt(sc.nextLine()); int noOfFullContainers = amountOfBricks/containerCapacity; int noOfTotalContainers; int noOfBlocksInNotFullContainers = amountOfBricks%containerCapacity; if(noOfFullContainers==0){ noOfTotalContainers = noOfFullContainers + 1; } else { noOfTotalContainers = noOfFullContainers; } System.out.println("No of full containers we have: "+noOfFullContainers); System.out.println("No of total containers we have: "+noOfTotalContainers); System.out.println("No of blocks in the container that is not completely full: "+noOfBlocksInNotFullContainers); } } //Output:How many Lego bricks do we have? Choose an odd number between 50 and 100: 75 How many Lego blocks fit in one container?

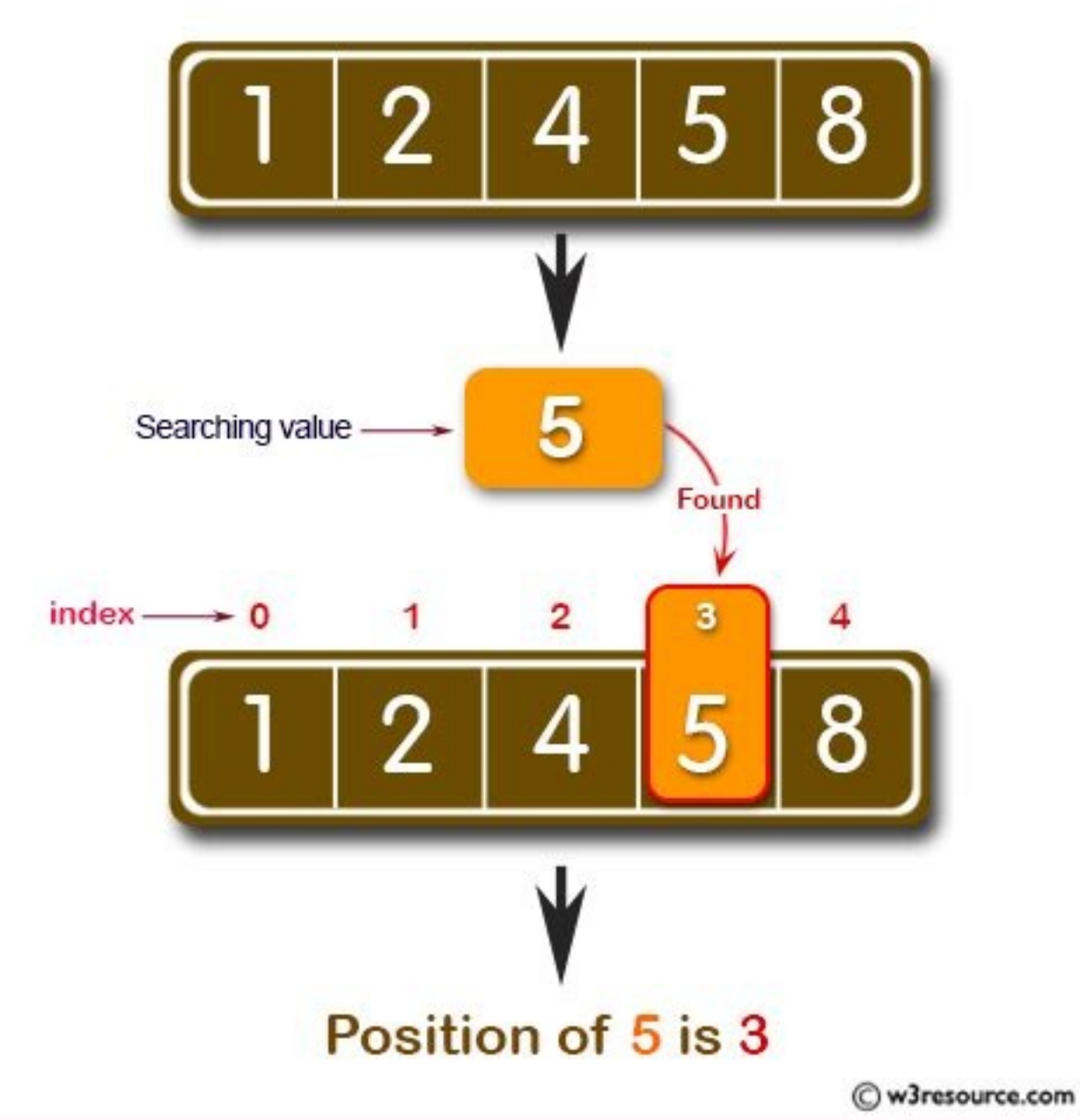
```



```

Choose an even number between 5 and 10: 8 No of full containers we have: 9 No of total containers we have: 10 No of blocks in the container that is not completely full: 3 Using the double types, implement the following: Suppose a product costs 9.99 net, calculate its gross value (we assume VAT of 23%). Then multiply it by 10,000 (i.e., we sold 10,000 copies of this product), and calculate this value excluding VAT. Implement the above actions using the Big Decimal class. Print on the console all computed values and compare their values. What conclusions do you have? import java.math.BigDecimal; class Main { public static void main(String[] args) { double netValue = 9.99; double VAT = 23.0; //calculating gross value double grossValue = netValue + (VAT*netValue/100); System.out.println("The gross value is: "+grossValue); // multiplying the value by 10000 double grossValue10000 = grossValue * 10000; System.out.println("The gross value for 10000 units is: "+grossValue10000); // calculating price excluding VAT 23% double grossValueExcludingVAT = grossValue10000 - (VAT*grossValue10000/100); System.out.println("The value for 10000 units excluding VAT is: "+grossValueExcludingVAT); // doing the same operations using BigDecimal instead of double System.out.println("Using BigDecimal instead of double---"); BigDecimal netValue big = new BigDecimal("9.99"); BigDecimal VAT big = new BigDecimal("23.0"); //calculating gross value using BigDecimal double grossValue10000BigDecimal = 10000 * (netValue + (VAT*netValue/100)); System.out.println("The gross value for 10000 units using BigDecimal is: "+grossValue10000BigDecimal); // calculating price excluding VAT 23% using BigDecimal double grossValueExcludingVAT10000BigDecimal = grossValue10000BigDecimal - (VAT*grossValue10000BigDecimal/100); System.out.println("The value for 10000 units excluding VAT is: "+grossValueExcludingVAT10000BigDecimal); // calculating price excluding VAT 23% using BigDecimal double grossValueExcludingVAT10000BigDecimal = grossValue10000BigDecimal.subtract(VAT.big.multiply(grossValue10000BigDecimal.divide(BigDecimal.valueOf(10000)))); System.out.println("The value for 10000 units excluding VAT is: "+grossValueExcludingVAT10000BigDecimal); } } Output: The gross value is: 12.2877000000000001 The gross value for 10000 units is: 122877.000000000001 The gross value for 10000 units excluding VAT is: 94615.290000000001 --- Using BigDecimal instead of double--- The gross value is: 12.2877 The gross value for 10000 units is: 122877.00000 The value for 10000 units excluding VAT is: 94615.2900000 The accuracy is higher when we use BigDecimal instead of double The user enters his height (in inches) and weight (in pounds). The variables passed by the user are assigned to the float type. After calculating the BMI value, the value will be assigned to the appropriate range and the correct message will appear on the console. You can use the if-else-if ladder for printing the message on the console. Intervals of BMI index: 16.00 or less = starvation 16.00-16.99 = emaciation 17.00-18.49 = underweight 18.50-22.99 = normal, low range 23.00-24.99 = normal, high range 25.00-26.99 = overweight, low range 27.00-29.99 = overweight, high range 30.00 or more = obesity 30.00-34.99 = overweight, high range 35.00 or more = obesity 35.00-39.99 = severe obesity 40.00 or more = morbid obesity. import java.util.Scanner; class Main { public static void main(String[] args) { Scanner sc = new Scanner(System.in); System.out.println("Enter the height (in inches): "); float height = sc.nextFloat(); System.out.println("Enter the height (in pounds): "); float weight = sc.nextFloat(); bmi = (float)(weight/ Math.pow(height, 2) * 703); System.out.println("The BMI is: "+bmi); if(bmi<16.00) System.out.println("starvation"); else if(bmi<16.00 & bmi <= 16.99) System.out.println("emaciation"); else if(bmi>=17.00 & bmi <= 18.49) System.out.println("underweight"); else if(bmi>=18.50 & bmi <= 22.99) System.out.println("normal, low range"); else if(bmi>=23.00 & bmi <= 24.99) System.out.println("normal, high range"); else if(bmi>=25.00 & bmi <= 26.99) System.out.println("overweight, low range"); else if(bmi>=27.00 & bmi <= 29.99) System.out.println("overweight, high range"); else if(bmi>=30.00 & bmi <= 34.99) System.out.println("1st degree obesity"); else if(bmi>=35.00 & bmi <= 39.99) System.out.println("2nd degree obesity"); else System.out.println("3rd degree obesity"); } } Output: Enter the height (in inches): 69 Enter the weight (in pounds): 150 The BMI is: 22.147878787878787 normal, low range Write a program that sums even numbers from 1 to 100 using for loop. class Main { public static void main(String[] args) { int sum=0; for(int i=1;i<=100;i++) { if(i%2==0) { sum = sum + i; } } System.out.println("The sum of even numbers from 1-100 is: "+sum); } } Output: The sum of even numbers from 1-100 is: 2550 Write a program that will use the while loop to find the largest and smallest number from the set of 10 randomly drawn integers from 0 to 100. In this task, do not use arrays or other collections. import java.util.Random; class Main { public static void main(String[] args) { int min = 100; int largest=0; int counter=0; while(counter<10) { int randomNum = new Random().nextInt(100); System.out.println("Generated random number is: "+randomNum); if(randomNum>largest) largest = randomNum; if(randomNum<min) min = randomNum; counter++; } System.out.println("Largest = "+largest); System.out.println("Smallest = "+min); } } Output: Generated random number is: 47 Generated random number is: 10 Generated random number is: 99 Generated random number is: 100 Generated random number is: 100 Generated random number is: 100 Generated random number is: 100 Generated random number is: 100 Generated random number is: 100 Generated random number is: 100 Largest = 100 Smallest = 100
repository with the exercises-solutions topic, visit your repo's landing page and select "manage topics". Learn more You can't perform that action at this time.

```

You signed in with another tab or window. Reload to refresh your session. You signed out in another tab or window. Reload to refresh your session.