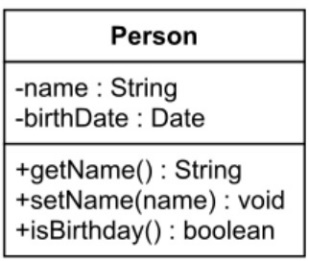
Names: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Use the UML class diagram presented below to identify the class name, attributes, and methods. Indicate the visibility of the attributes and methods.



Class name: Person

Attributes: String name – private, Date birthdate – private

Methods: getName() , setName(name), isBirthday() all methods are public

1. Use the Javadocs descriptions provided in the code below to implement the methods and make the correct calls in main.

import java.util.Scanner;  
  
public class WSWeek3 {  
 */\*\*  
 \* Main method used to call other methods  
 \* @parameter String args[]  
 \*/* public static void main(String args[]){  
 *//Creates a Scanner object named in by calling the Scanner*

*//constructor and passing System.in as the parameter* Scanner in = new Scanner(System.*in*);  
 *//call maxMin method passing in as a parameter  
 minMax*(in);  
 *//since factorial receives a parameter, we need to read an int  
 //number to pass as the parameter  
 //if the number is negative, we transform to a positive  
 //since factorial returns a value, we need to call it  
 //inside a System.out.println or store it in a variable.  
 //int fac = factorial(num);* System.*out*.println("Enter a number:");  
 int num = in.nextInt();  
 if(num < 0)  
 num \*= -1;  
 System.*out*.printf("%d! = %d\n", num, *factorial*(num));  
  
 *//read the number to call perfect  
 //since perfect return true or false, you need to have  
 //an if to test and print the correct message* System.*out*.print("Enter a number to verify perfect number: ");  
 num = in.nextInt();  
 if(*perfect*(num))  
 System.*out*.printf("%d is a perfect number\n", num);  
 else  
 System.*out*.printf("%d is NOT a perfect number \n", num);  
 }  
 */\*\*  
 \* minMax method  
 \* Asks the user to enter the total number to be read.  
 \* Determines and print the min and max of the numbers read.  
 \* @param in  
 \*/  
 // Possible algorithm to solve that problem:  
 // 1 – declare your variables and initialize them  
 // 2 - read the total of number that will be entered  
 // 3 – have a loop that starts on 0 and do until < total  
 // 4 – inside the for loop  
 // 5 – read a number  
 // 6 – if the number is the first read – initialize min and max with  
 // that number  
 // 7 – if num < min – update min  
 // 8 – if num > max – update max  
 // 9 – prints the min and max outside of the loop* public static void minMax(Scanner in){  
 int total = 0, min = 0, max =0;  
 System.*out*.print("Enter the amount of numbers that will be read: ");  
 total = in.nextInt();  
 for(int i = 0; i < total; i++){  
 System.*out*.print("Enter a number: ");  
 int num = in.nextInt();  
 if(i == 0){  
 min = num;  
 max = num;  
 }  
 if(num < min)  
 min = num;  
 if(num > max)  
 max = num;  
 }  
 System.*out*.printf("Min: %d Max: %d\n", min, max);  
 }  
  
 */\*\*  
 \* factorial method  
 \* calculates the factorial of a number  
 \* @param num - number to calculate the factorial  
 \* @return factorial of num  
 \*/  
 //Possible algorithm to solve this problem:  
 //1 - test if num == 0 return 1  
 //2 - declare and initialize fat variable with num  
 //3 - use a while loop num > 1  
 //4 - update num, by decrementing 1  
 //5 - fat = fat \* num or fat \*= num  
 //6 - out of the loop - return fat* public static int factorial(int num){  
 if(num == 0) return 1;  
 int fat = num;  
 while(num > 1){  
 num--;  
 fat = fat \* num;  
 }  
 return fat;  
 }  
  
 */\*\*  
 \* perfectNumber method  
 \* verifies if a number is a perfect number.  
 \* A perfect number is a positive integer that is equal to the  
 \* sum of its positive divisors, excluding the number itself.  
 \* For instance, 6 has divisors 1, 2 and 3 (excluding itself),  
 \* and 1 + 2 + 3 = 6, so 6 is a perfect number.  
 \* @param num  
 \* @return boolean - true if num is a perfect number, false otherwise  
 \*/  
 //Possible algorithm to solve this problem:  
 // 1 - if num is zero, return false  
 // 2 - if num is negative, transform to a positive number  
 // 3 - initialize your div variable with 1 (first divisor)  
 // 4 - do a loop starting on 2 and going <= num/2 (which is the second great divisor of the number)  
 // 5 - inside the loop test if num % i == 0 add i to div, since i is a divisor of num  
 // 6 - outside the loop test if div == num return true, return false otherwise* public static boolean perfect(int num){  
 if(num == 0) return false;  
 if(num < 0) num \*= -1;  
 int div = 1;  
 for(int i = 2; i <= num/2; i++){  
 if(num % i == 0)  
 div = div + i;  
 }  
 if(div == num)  
 return true;  
 return false;  
 }  
}