Problem 11: The Planet Explorer

You are a scientist at a space research center. The center is currently researching various planets and their properties. One of the properties they are interested in is the surface area of these planets. Considering that a planet can be approximated as a sphere, your task is to create a method that calculates the surface area of a sphere given its radius.

Hint: The formula to calculate the surface area of a sphere is:

A = 4πr^2

Where:

- A is the surface area of the sphere
- r is the radius of the sphere

Class Definition:

```
public class PlanetExplorer {
public double calculateSurfaceArea(double radius);
}
```

Inputs:

The method calculateSurfaceArea(double radius) will receive one parameter:

- radius : a double representing the radius of the sphere (planet).

Outputs:

The method will return a double - the surface area of the sphere.

Example:

Sample Input:

```
PlanetExplorer explorer = new PlanetExplorer();
explorer.calculateSurfaceArea(3.0);
```

Sample Output:

113.10

Note:

In the sample input, the radius of the planet (sphere) is 3.0 units. The surface area is 4π*3^2 = 113.10, so the method returns 113.10 as the output. Your method will be essential in helping the scientists at the space research center in understanding the properties of various planets. Good luck, scientist!

Program:

```
package ClassProgram;
public class PlanetExplorer {
	public double calculateSurfaceArea(double radius) {
		return 4*Math.PI*radius*radius;
	}
}
package ClassProgram;
import java.util.Scanner;
public class PlanetExplorerApp {
	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		double radius=scan.nextDouble();
		PlanetExplorer explorer=new PlanetExplorer();
		System.out.printf("%.2f",explorer.calculateSurfaceArea(radius));
		scan.close();
	}
}
```

Output:

```
3
113.10
```

Problem 12: The Height Converter

You are part of a sports data management team. The team is developing a new feature for their application where the heights of players, currently recorded in inches, need to be displayed in feet for an international audience. Your task is to create a method that takes a height given in inches and converts it into feet.

Hint: The conversion factor from inches to feet is 1 foot = 12 inches.

Class Definition:

public class HeightConverter {

public double convertInchesToFeet(double inches);

}

Inputs:

The method convertInchesToFeet(double inches) will receive one parameter:

- inches ($0 \leq inches \leq 10^9$): a double representing the height in inches.

Outputs:

The method will return a double - the height converted to feet.

Example:

Sample Input:

HeightConverter converter = new HeightConverter();

converter.convertInchesToFeet(72.0);

Sample Output:

6.00

Note:

In the sample input, the height of the player is 72.0 inches. The height in feet is 72/12 = 6.00, so the method returns 6.00 as the output. Your method will be crucial in helping the sports data management team present the data in a format familiar to the international audience. Good luck, data manager!

Program:

```
package ClassProgram;
public class HeightConverter {
	public double convertInchesToFeet(double inches) {
		return inches/12.0;
	}
}
package ClassProgram;
import java.util.Scanner;
public class HeightConverterApp {
	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		double inches=scan.nextDouble();
		HeightConverter converter=new HeightConverter();
		System.out.printf("%.2f",converter.convertInchesToFeet(inches));
		scan.close();
	}
}
```

Output:

72

6.00

Problem 13: The Finance Calculator

You are a software developer at a financial technology company. The company is building a new feature in their app that calculates the simple interest for users wanting to take out loans. Your task is to create a method that calculates the simple interest given the principal amount, rate of interest, and time.

Hint: The formula to calculate simple interest is:

I = P * R * T

Where:

- I is the simple interest
- P is the principal amount
- R is the rate of interest (in decimal)
- T is the time (in years)

Class Definition:

public class FinanceCalculator {

public double calculateSimpleInterest(double principal, double rate, double time);

}

Inputs:

The method calculateSimpleInterest(double principal, double rate, double time) will receive three parameters:

- principal (1 ≤ principal ≤ 10^9): a double representing the principal amount.
- rate (0 ≤ rate ≤ 1): a double representing the rate of interest in decimal.
- time (1 ≤ time ≤ 10^5): a double representing the time in years.

Outputs:

The method will return a double - the simple interest.

Example:

Sample Input:

FinanceCalculator calculator = new FinanceCalculator();

calculator.calculateSimpleInterest(1000.0, 0.05, 2.0);

Sample Output:

100.00

Note:

In the sample input, the principal amount is 1000.0 units, the rate of interest is 0.05, and the time is 2.0 years. The simple interest is 1000.0 * 0.05 * 2.0 = 100.00, so the

method returns 100.00 as the output. Your method will be crucial in helping users plan their finances. Good luck, developer!

Program:

```
package ClassProgram;
public class FinanceCalculator {
	public double calculateSimpleInterest(double principal, double rate, double time) {
		return principal*rate*time;
	}
}
package ClassProgram;
import java.util.Scanner;
public class FinanceCalculatorApp {
	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		double principal=scan.nextDouble();
		double rate=scan.nextDouble();
		double time=scan.nextDouble();
		FinanceCalculator calculator = new FinanceCalculator();
	System.out.printf("%.2f",calculator.calculateSimpleInterest(principal,rate,time));
		scan.close();
	}
}
```

Output:

1000

0.05

2.0

100.00

Problem 14: Time Converter

You are developing a time tracking application for a company. The employees log their time in minutes. The management wants to see these durations in hours for better understanding. Your task is to write a function that can convert minutes into hours.

Function:

public static double convertToHours(int minutes);

Inputs:

The function convertToHours(int minutes) will receive one parameter:

- minutes (1 ≤ minutes ≤ 10^6): an integer which represents the number of minutes to be converted.

Outputs:

The function will return a double - the equivalent number of hours.

Example:

Sample Input:

convertToHours(90);

Sample Output:

1.5

Note:

In the sample input, the number of minutes given to the function is 90. The equivalent in hours is 90/60=1.5, so the function returns 1.5 as the output. Your function should work accurately to provide correct information to the management.

Program:

```
package ClassProgram;
import java.util.Scanner;
```

```
public class TimeConverter {
	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		int minutes=scan.nextInt();
		System.out.printf("%.2f",convertToHours(minutes));
		scan.close();
	}
	public static double convertToHours(int minutes) {
		return minutes/60.0;
	}
}
```

Output:

90

1.50

Problem 15: Halve It

You are helping a friend in developing a financial app. The app has a feature where it calculates half of the entered amount for splitting bills. Your task is to write a function that takes a number and returns its half.

Function:

public static double halveTheNumber(double num);

Inputs:

The function halveTheNumber(double num) will receive one parameter:

- num (0 ≤ num ≤ 10^9): a double which represents the amount entered by the user to be halved.

Outputs:

The function will return a double - the result of halving the num.

Example:

Sample Input:

halveTheNumber(150.00);

Sample Output:

75.00

Note:

In the sample input, the number given to the function is 150.00. The half of this number is 150.00/2=75.00, so the function returns 75.00 as the output. Make sure your function works correctly to split the bills accurately.

Program:

```
package ClassProgram;
import java.util.Scanner;
public class HalveIt {
	public static void main(String[] args) {
		Scanner scan=new Scanner(System.in);
		double num=scan.nextInt();
		System.out.printf("%.2f",halveTheNumber(num));
		scan.close();
	}
	public static double halveTheNumber(double num) {
		return num/2.0;
	}
}
```

Output:

150

75.00