Module case_tools
Package assignments

Class SmartCalculator

java.lang.Object
    assignments.SmartCalculator

```
public class SmartCalculator
extends Object
```

Constructor Summary

Constructors

Constructor	Description
`SmartCalculator()`	Constructs a new instance of the SmartCalculator class with default values.
`SmartCalculator(double principal, double rate, double time, double totalMarks, double obtainedMarks)`	

Method Summary

All Methods Static Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
double	compoundInterest()	Calculates the compound interest based on the principal amount, interest rate, and time.
double	factorial(int n)	Calculates the factorial of a given integer.
double	getAmount()	Retrieves the calculated amount stored in the SmartCalculator instance.
double	getMarks()	Retrieves the marks obtained stored in the SmartCalculator instance.
double	getMaxMarks()	Retrieves the maximum marks stored in the SmartCalculator instance.
double	getPrincipal()	Retrieves the principal amount stored in a smartCalculator instance.
double	getRate()	.
double	getTime()	Retrieves the time period stored in a SmartCalculator instance.
static void	main(String [] args)	The main method of the SmartCalculator program allows users to perform various calculations using the SmartCalculator class.
double	mean(double[] numbers)	Calculates the mean (average) of an array of numbers.
double	percentage()	Calculates the percentage based on the marks obtained and maximum marks.
void	setAmount(double amount)	Sets the calculated amount in a smartCalculator interest.
void	setMarks(double obtainedMarks)	Sets the marks obtained for calculations in the SmartCalculator instance.
void	setMaxMarks(double totalMarks)	Sets the maximum marks for calculations in the SmartCalculator instance.
void	setPrincipal(double principal)	Sets the principal amount for calculating the simple interest.
void	setTime(double time)	Sets the time period in a smartCalaculator instance.
double	simpleInterest()	Calculates the simple interest based on the principal amount, interest rate, and time.

Methods inherited from class java.lang.Object

`equals` , `getClass` , `hashCode` , `notify` , `notifyAll` , `toString` , `wait` , `wait` , `wait`

Constructor Details

SmartCalculator

```
public SmartCalculator()
```

Constructs a new instance of the SmartCalculator class with default values. This non-parameterized constructor initializes the instance variables for principal, rate, time, amount, maximum marks, and marks obtained to default values 0.

SmartCalculator

```
public SmartCalculator(double principal,
                       double rate,
                       double time,
                       double totalMarks,
                       double obtainedMarks)
```

Parameters:

`principal` - The starting amount of money

`rate` - The percentage of interest charged on the principal amount over a certain period

`time` - Time refers to the duration for which the principal amount is borrowed or invested

`totalMarks` - The total marks

`obtainedMarks` - The total marks obtained

Method Details

getPrincipal

```
public double getPrincipal()
```

Retrieves the principal amount stored in a smartCalculator instance.

Returns:

The principal amount as a double.

getRate

```
public double getRate()
```

. This method returns the value of the interest rate attribute of a smartcalculator instance.

Returns:

The interest rate used in the operation.

setPrincipal

```
public void setPrincipal(double principal)
```

Sets the principal amount for calculating the simple interest. This method updates the principal amount attribute with the specified value.

Parameters:

`principal` - The principal amount to be set for calculating the simple interest.

getTime

```
public double getTime()
```

Retrieves the time period stored in a SmartCalculator instance.

Returns:

The time period as a double.

setTime

```
public void setTime(double time)
```

Sets the time period in a smartCalaculator instance.

Parameters:

`time` - The time period to be set as a double.

getAmount

```
public double getAmount()
```

Retrieves the calculated amount stored in the SmartCalculator instance.

Returns:

The calculated amount as a double.

setAmount

```
public void setAmount(double amount)
```

Sets the calculated amount in a smartCalculator interest.

Parameters:

`amount` - The calculated amount to be set as a double.

getMaxMarks

```
public double getMaxMarks()
```

Retrieves the maximum marks stored in the SmartCalculator instance.

Returns:

The maximum marks as a double.

setMaxMarks

```
public void setMaxMarks(double totalMarks)
```

Sets the maximum marks for calculations in the SmartCalculator instance.

Parameters:

`totalMarks` - The total marks to be set as the maximum marks, specified as a double.

getMarks

```
public double getMarks()
```

Retrieves the marks obtained stored in the SmartCalculator instance.

Returns:

The marks obtained as a double.

setMarks

```
public void setMarks(double obtainedMarks)
```

Sets the marks obtained for calculations in the SmartCalculator instance.

Parameters:

`obtainedMarks` - The marks obtained to be set, specified as a double.

simpleInterest

```
public double simpleInterest()
```

Calculates the simple interest based on the principal amount, interest rate, and time. This method uses the simple interest formula: amount = (principal * rate * time) / 100, where: - amount is the total interest accrued, - principal is the initial principal amount, - rate is the interest rate per period, - time is the number of periods.

Returns:

The calculated simple interest.

compoundInterest

```
public double compoundInterest()
```

Calculates the compound interest based on the principal amount, interest rate, and time. This method uses the compound interest formula: amount = principal * Math.pow((1 + rate), time), where: - amount is the total amount after compounding, - principal is the initial principal amount, - rate is the interest rate per period, - time is the number of periods. The compound interest is then calculated by subtracting the principal amount from the calculated amount after compounding.

Returns:

The calculated compound interest.

mean

```
public double mean(double[] numbers)
```

Calculates the mean (average) of an array of numbers.

Parameters:

`numbers` - an array of double values for which the mean is calculated

Returns:

the mean (average) of the numbers in the array

factorial

```
public double factorial(int n)
```

Calculates the factorial of a given integer. This method takes a non-negative integer n as input and returns its factorial. The factorial of a non-negative integer is the product of all positive integers less than or equal to n.

Parameters:

`n` - The integer for which the factorial is to be calculated.

Returns:

The factorial of the input integer.

percentage

```
public double percentage()
```

Calculates the percentage based on the marks obtained and maximum marks.

Returns:

The calculated percentage.

main

```
public static void main(String [] args)
```

The main method of the SmartCalculator program allows users to perform various calculations using the SmartCalculator class. It presents a menu with options to calculate Simple Interest, Compound Interest, Mean, Factorial, Percentage, or exit the program. Users can input values based on their selected option, and the program will display the calculated result.