Sirma

e: office@sirma.bg
m: +359 2 9768310
a: 135 Tsarigradsko Shosse, blvd.
1784 Sofia, Bulgaria

Prototype & Inheritance - Exercise

In this exercise, you will practice Prototype & Inheritance

1. Person

Write a program that takes firstName and lastName as parameters and returns an object with firstName, lastName, and **fullName** properties. If firstName or lastName changes, fullName should also change. If fullName changes to a valid format, firstName and lastName should change accordingly.

2. Person and Student

Create a **Person** class with name and age properties, and a **Student** class that extends Person and adds a school property.

3. Extend Prototype

Write a function that receives a class and attaches a species property with the value "Human" and a toSpeciesString() method that returns a string in the format "I am a <species>. <toString()>".

4. Class Hierarchy

Write a function that returns three classes: Figure, Circle, and Rectangle. Figure has a units property (default "cm"), a getter area, and methods changeUnits and toString. Circle and Rectangle extend Figure and override area and toString appropriately.

5. Shapes and Area Calculation

Create a class hierarchy where Shape is the base class with a calculateArea method. Create Triangle, Square, and Circle classes that inherit from Shape and implement their own calculateArea methods.

6. Vehicle Inheritance

Create a base class Vehicle with properties make and model. Create Car and Bike classes that extend Vehicle and add properties specific to each type (e.g., doors for Car, type for Bike).

7. Employee Hierarchy

e: office@sirma.bg
m: +359 2 9768310
a: 135 Tsarigradsko Shosse, blvd.
1784 Sofia, Bulgaria

Create a base class Employee with properties name and salary. Extend it with classes Manager and Engineer. Add methods to calculate bonuses based on different criteria for managers and engineers.

8. Animal Inheritance

Create an Animal class with properties name and age, and methods eat and sleep. Extend it with Dog and Cat classes, adding properties and methods specific to dogs and cats (e.g., bark for Dog, meow for Cat).

9. Library System

Create a LibraryItem class with properties title and author. Extend it with Book and Magazine classes. Add methods to display details specific to books and magazines.

10. Array Extension

Extend the built-in Array object with additional functionality. Implement the following methods:

- first(): returns the first element of the array.
- skip(n): returns a new array excluding the first n elements.
- take(n): returns a new array containing the first n elements.
- product(): returns the product of all array elements.
- unique(): returns a new array with unique elements.

Structure your code as an IIFE to add these methods to the Array prototype.

11. String Extension

Extend the built-in String object with additional functionality. Implement the following methods:

- ensureStart(str): ensures the current string starts with the given str.
- ensureEnd(str): ensures the current string ends with the given str.
- isEmpty(): returns true if the string is empty, false otherwise.
- capitalize(): returns the string with the first letter capitalized.
- truncateWords(n): returns the string truncated to n words, appending an ellipsis if necessary.

12. Extensible Object

Create an object that can clone the functionality of another object into itself. Implement an extend(template) method that copies all properties from the template to the parent object. If a property is a function, add it to the object's prototype instead.

13. Company

e: office@sirma.bg
m: +359 2 9768310
a: 135 Tsarigradsko Shosse, blvd.
1784 Sofia, Bulgaria

Create a class hierarchy representing employees in a company. Create an abstract Employee class with properties name, age, salary, and tasks. Subclasses Junior, Senior, and Manager should add specific properties and methods, including work() and getSalary().

14. Musical Instruments

Create a class hierarchy for musical instruments:

- Instrument: properties name and type, methods play() and tune().
- Subclass StringInstrument: additional property numberOfStrings, override tune() to print a specific message for string instruments.
- Subclass PercussionInstrument: additional property drumSize, override tune() to print a specific message for percussion instruments.

15. Smart Devices

Create a class hierarchy for smart devices:

- Device: properties brand, model, batteryLife, and methods charge() and turnOn().
- Subclass Smartphone: additional properties screenSize and os, override charge() to include charging time for smartphones.
- Subclass Smartwatch: additional properties strapMaterial and waterResistance, override charge() to include charging time for smartwatches.

16. Bank System

Create a class hierarchy for a bank system:

- BankAccount: properties accountNumber, balance, methods deposit(amount), withdraw(amount), and getBalance().
- Subclass SavingsAccount: additional property interestRate, method applyInterest().
- Subclass CheckingAccount: additional property overdraftLimit, override withdraw(amount) to account for overdraft.

17. E-commerce

Create a class hierarchy for an e-commerce system:

- Product: properties id, name, price, and method applyDiscount(discount).
- Subclass Electronics: additional properties warrantyPeriod and brand.
- Subclass Clothing: additional properties size and material.

18. School System

Create a class hierarchy for a school system:

e: office@sirma.bg
m: +359 2 9768310
a: 135 Tsarigradsko Shosse, blvd.
1784 Sofia, Bulgaria

- Person: properties name and age.
- Subclass Student: additional properties grade and school, method study().
- Subclass Teacher: additional properties subject and salary, method teach().