Лабораторная работа №8

Контрольные вопросы

1) Что такое классы объектов с точки зрения Javascript?
2) Какие свойства называются общими, а какие собственными?
3) Что такое прототипы и где они хранятся в структуре объекта? Что такое цепочка прототипов?
4) Как реализуется наследование в Javascript? Что происходит при обращении ко свойству объекта являющимся наследником других объектов?
5) Как можно создать объект наследующих прототип, получить прототип объекта и заменить прототип объекта?
6) Каков механизм чтения и записи унаследованных свойств?
7) Что такое конструктор? Где конструктор хранит прототип объекта?
8) Для чего нужен оператор instanceof? Что такое свойство constructor?
9) Что такое флаги и дескрипторы свойств?
10) Какие методы есть по работе с дескрипторами?
11) Какие методы есть для ограничения доступа к объекту?
12) Какие методы есть для проверки состояния доступа к объекту?
13) Что такое аксессоры и как их можно задавать?
14) Что такое class-синтаксис и каков его формат?
15) Как осуществляется наследование и для чего нужно ключевое слово super в class-синтаксисе?
16) Как задаются статические и приватные свойства и методы в class-синтаксисе?

Задание

Разработать класс объектов по вариантам задания (через прототипы, без использования **class**-определения). Каждый класс из вариантов должен иметь переопределенный метод toString() для вывода состояния объекта в консоль, а также наследоваться от родительского класса BaseObject, который должен иметь методы:

- регистрации факта, времени и аргументов вызова методов дочерних классов (т.е. дочерний класс должен регистрировать эту информацию при вызове своих методов в своём родительском классе).
- очистку списка регистрации действий.
- вывод списка зарегистрированных действий в консоль.

Интерфейс программы должен содержать возможность создавать объекты класса и выполнять с ними все возможные действия, с выводом результатов на экран.

Варианты:

1. «**Комплексное число**» – **Complex**. Разработать класс комплексных чисел. Класс должен содержать методы для изменения и получения значения действительной и мнимой части, для реализации операций сложения, вычитания, умножения, деления, присваивания комплексных чисел.

2. «**Дробь**» – **Fraction.** Разработать класс в виде пары целых положительных чисел (*m,n*) а также отдельно знак дроби. Класс должен содержать методы для изменения и получения значения числителя и знаменателя, сложения, вычитания, умножения, деления и присваивания дробей.

3. «**Вектор**» – **Vector.** Разработать класс вектора размерности *n*. Реализовать методы для изменения и получения значения компонента вектора, вычисления длины вектора, скалярного произведения, сложения, умножения, умножения на скаляр.

4. «**Квадратная матрица**» – **Matrix**. Разработать класс квадратной матрицы *n x n*. Реализовать методы для изменения и получения значения элемента матрицы, сложения, вычитания, умножения матриц; вычисления индексов максимального и минимального элемента матрицы.

5. **«Многочлен» – Polynom.** Разработать класс полинома степени *n*. Реализовать методы для изменения и получения значения указанного коэффициента, вычисления значения полинома; сложения, вычитания, умножения полиномов.

6. **«Фигуры» – Shapes.** Разработать класс для описания плоских фигур: круг, прямоугольник, треугольник. Включить методы для получения и изменения параметров фигур, перемещения на плоскости, вращения, нахождения площади и периметра фигуры.

7. «**Множество целых чисел**» –**Set.** Разработать класс множества целых чисел мощности *n*. Реализовать методы для определения принадлежности заданного элемента множеству, добавление\удаление элемента, пересечения, объединения, разности двух множеств.

8. «**Массив строк**» – **StringArray.** Разработать класс для представления массива строк. Реализовать методы для добавления\удаления строк, для поэлементной конкатенации двух массивов, упорядочения строк по длине, слияния двух массивов строк с удалением повторяющихся строк, а также формирование массива количества слов в каждой строке.

9. «**Массив бит**» – **BitArray.** Разработать класс представляющий собой массив битов длины n. Реализовать методы для установки и получения значения бита на заданной позиции, изменения размера массива (справа и слева), сдвиг битов вправо\влево на заданное число позиций, битовые операции and и or для двух массивов.

10. «**Булева матрица»** – **BoolMatrix.** Разработать класс представляющий собой матрицу булевых значений размерности *n*×*m*. Реализовать методы для изменения и получения значения указанного элемента, логического сложения, умножения и инверсии матриц. Реализовать метод для подсчета количества true и false значений в матрице.

11. «**Односвязный список»** – **LinkedList.** Разработать класс для работы с односвязным списком с целыми числами. Реализовать методы добавления элемента на заданную позицию, удаление всех элементов с заданным значением, получение значения по заданному индексу, объединение двух списков, разделение списка на два с указанной позиции, реверс списка.

12. «**Бинарное дерево»** – **BinaryTree.** Разработать класс для работы с бинарным деревом, узлы которого содержат натуральные числа. Реализовать методы добавления и удаления узлов, получения массива узлов с заданным значением, определения высоты и количества листьев у дерева.