

1 Bag of Dice

Assume you have an implementation of the Dice class from lecture. The class has the following public methods:

- public Dice(int numSides)
- public void roll()
- public int getTop()
- public int getSides()
- public void setTop(int faceValue)

In some games, several dice are rolled at once, which leads to interesting non-uniform probability distributions. So now we will create a bag of dice, a class that holds several dice and has a method to roll all of them simultaneously.

```
class BagOfDice {
    // this array will hold all dice
    private Dice[] dice;

    /** Create a new bag of dice with given number of dice.
     * The number of sides will be 6 for all new dice. */
    public BagOfDice(int numDice) {

    }

    /** Create a new bag of dice with given numbers of sides.
     * The number of dice should be equal to the array length. */
    public BagOfDice(int[] numSides) {

    }

    /** returns the array of Dice objects being used by this class */
    public Dice[] getDice() {

    }

    /** Roll dice and return the sum of numbers on top. */
    public int rollAllDice() {

    }
}
```

2 Polynomials

In this part, we will create a class that represents a polynomial as an array of coefficients. Then we will extend the polynomial class to implement a quadratic.

```
class Polynomial {
    // the array of coefficients coeffs[i] is the coefficient next to x^i
    protected double[] coeffs;

    /** Create a new polynomial with the given coefficients. */
    public Polynomial(double[] coefficients) {

    }

    /** =The degree of this polynomial (the degree of the
     * highest nonzero term.) */
    public int getDegree() {

    }

    /** ={This Polynomial is the same as Polynomial p, true or false} */
    public boolean equals(Polynomial p) {

    }

    /* = The value of this polynomial evaluated at x */
    public double evaluate(double x) {

    }
}

class Quadratic _____ Polynomial {
    /** Construct a polynomial  $ax^2 + bx + c$ . */
    public Quadratic(double a, double b, double c) {

    }

    /** =The roots of this quadratic. The returned array
     * will be of size 2 or 1, or null if roots are complex. */
    public double[] getRealRoots() {

    }
}
```

If you have time (or after section), try to implement new instance methods into the Polynomial class to add another polynomial to *this* one and to compute the derivative. What method headers would be appropriate?