Student Name: Bogdan Itsam Dorantes-Nikolaev 042101002
Class and Section COMP 201: Data Structures and Algorithms Section-2

Project: The MyInteger Class

Problem Description:
Design a class named MyInteger. The class contains:

- An int data field named value that stores the int value represented by this object.
- A constructor that creates a MyInteger object for the specified int value.
- A get method that returns the int value.
- Methods isEven(), isOdd(), and isPrime() that return true if the value is even, odd, or prime, respectively.
- Static methods isEven(int), isOdd(int), and isPrime(int) that return true if the specified value is even, odd, or prime, respectively.
- Static methods isEven(MyInteger), isOdd(MyInteger), and isPrime(MyInteger) that return true if the specified value is even, odd, or prime, respectively.
- Methods equals(int) and equals(MyInteger) that return true if the value in the object is equal to the specified value.
- A static method parseInt(**char**[]) that converts an array of numeric characters to an int value.
- A static method parseInt(String) that converts a string into an int value.

  Draw the UML diagram for the class. Implement the class. Write a client program that tests all methods in the class.

Design:

```
// By Bogdan Itsam Dorantes-Nikolaev, COMP201, Instructor: Prof Adem Karahoca (October 16th, 2022)

// A class named MyInteger
class MyInteger {

// An int data field named value that stores the int value represented by this object.
   private int value;

// A constructor that creates a MyInteger object for the specified int value.
   public MyInteger(int inValue) {
      value = inValue;
   }

// A get method that returns the int value.
   public int getValue() {
      return value;
   }

// Methods isEven(), isOdd(), and isPrime() that return true if the value is even, odd, or prime, respectively.
   public boolean isEven() {
      return (value % 2) == 0;
   }

   public boolean isOdd() {
      return (value % 2) == 1;
   }

   public boolean isPrime() {
      if (value == 1 || value == 2) {
         return true;
      }
      else {
         for (int i = 2; i < value; i++) {
            if (i % value == 0) return false;
         }
      }
      return true;
   }

// Static methods isEven(int), isOdd(int), and isPrime(int) that return true if the specified value is even, odd, or prime, respectively.
   public static boolean isEven(int myInt) {
      return (myInt % 2) == 0;
   }

   public static boolean isOdd(int myInt) {
      return (myInt % 2) == 1;
   }

   public static boolean isPrime(int myInt) {
      if (myInt == 1 || myInt == 2) {
         return true;
      }
      else {
         for (int i = 2; i < myInt; i++) {
            if (i % myInt == 0) return false;
         }
      }
      return true;
   }

// Static methods isEven(MyInteger), isOdd(MyInteger), and isPrime(MyInteger) that return true if the specified value is even, odd, or prime, respectively.
   public static boolean isEven(MyInteger myInt) {
      return myInt.isEven();
   }

   public static boolean isOdd(MyInteger myInt) {
      return myInt.isOdd();
   }

   public static boolean isPrime(MyInteger myInt) {
      return myInt.isPrime();
   }

// Methods equals(int) and equals(MyInteger) that return true if the value in the object is equal to the specified value.
   public boolean equals(int testInt) {
      if (testInt == value)
         return true;
      return false;
   }

   public boolean equals(MyInteger myInt) {
      if (myInt.value == this.value)
         return true;
      return false;
   }

// A static method parseInt(char[]) that converts an array of numeric characters to an int value.
   public static int parseInt(char[] values) {
      int sum = 0;
      for (char i : values) {
         sum += Character.getNumericValue(i);
      }
      return sum;
   }

// A static method parseInt(String) that converts a string into an int value.
   public static int parseInt(String value) {
      return Integer.parseInt(value);
   }
}
```

Draw the UML class diagram here: