```java
// This method takes an array of Bag objects and a filename as input parameters
    static void save(Bag[] bags, String filename) {
        try {
            // Create a FileWriter object to write to the file specified by the filename
            // parameter
            FileWriter file = new FileWriter(filename);

            // Write the number of Bag objects in the array to the file
            file.write(bags.length + "\n");

            // Create an ExecutorService to execute each Bag object's write operation in a
            // separate thread to improve performance
            ExecutorService ex = Executors.newFixedThreadPool(5);

            // Create a Lock object to ensure that only one thread can write to the file at
            // a time
            Lock lock = new ReentrantLock();

            // Loop over each Bag object in the array
            for (Bag bag : bags) {
                // Submit a new Runnable to the ExecutorService to write the Bag object to the
                // file
                ex.submit(
                        () -> {
                            // Lock the Lock object to ensure that only one thread can write to the file at
                            // a time
                            lock.lock();
                            try {
                                // Write the Bag object's width, height, depth, hasWheels attribute, and color
                                // as a RGB integer value to the file
                                file.write(bag.width + " " + bag.height + " " + bag.depth + " " + bag.hasWheels + " "
                                        + bag.color.getRGB() + "\n");
                            } catch (Exception e) {
                                System.err.println(e);
                            } finally {
                                // Unlock the Lock object to allow other threads to write to the file
                                lock.unlock();
```

```
                        }
                    });
        }

        // Shut down the ExecutorService and wait for all threads to finish writing to
        // the file
        ex.shutdown();
        while (!ex.isTerminated())
            ;

        // Close the FileWriter object
        file.close();
    } catch (Exception e) {
        System.err.println(e);
    }
}

// This method takes a filename as an input parameter
static Bag[] open(String filename) {
    try {
        // Check that the filename has a valid extension (.txt or .bag)
        if (!filename.endsWith(".txt") && !filename.endsWith(".bag")) {
            throw new InvalidExtension(filename + " is not a valid name.");
        }

        // Create a FileReader object to read from the file specified by the filename
        // parameter
        FileReader file = new FileReader(filename);

        // Create a Scanner object to read from the FileReader object
        Scanner input = new Scanner(file);

        // Read the number of Bag objects in the file
        int nBags = Integer.valueOf(input.nextLine());

        // Create an array of Bag objects of size nBags
        Bag[] bags = new Bag[nBags];
```

```
            // Loop over each Bag object in the file
            for (int i = 0; i < nBags; i++) {
                // Read the Bag object's width, height, depth, hasWheels attribute, and color as
                // a RGB integer value from the file
                float width = input.nextFloat();
                float height = input.nextFloat();
                float depth = input.nextFloat();
                Boolean hasWheels = input.nextBoolean();
                Color color = new Color(Integer.valueOf(input.nextLine().trim()));

                // Create a new Bag object with the read attributes and add it to the array
                bags[i] = new Bag(width, height, depth, color, hasWheels);
            }

            // Close the FileReader and Scanner objects
            file.close();
            input.close();

            // Return the array of Bag objects read from the file
            return bags;
        } catch (Exception e) {
            System.err.println(e);

            // Return an empty array if an exception occurs
            return new Bag[0];
        }
    }

public static void main(String[] args) {
        Bag.save(new Bag[] {
                new Bag(10, 10, 10, Color.black, false),
                new Bag(5, 10.5f, 10, Color.white, false),
                new Bag(18, 8, 164, Color.blue, true),
        }, "output.txt");
        Bag[] bags = Bag.open("output.test");// this will print an error
        bags = Bag.open("output.txt");
```

```
        // it will have the data but id doesn'yt have to be the same order because we wrote using threads not normal
procedural code.
        for (Bag bag : bags) {
            System.out.println(bag);
        }
        //Bag [width=10.0, height=10.0, depth=10.0, color=java.awt.Color[r=0,g=0,b=0], hasWheels=false]
        //Bag [width=18.0, height=8.0, depth=164.0, color=java.awt.Color[r=0,g=0,b=255], hasWheels=true]
        //Bag [width=5.0, height=10.5, depth=10.0, color=java.awt.Color[r=255,g=255,b=255], hasWheels=false]
    }
```