MONGO DB

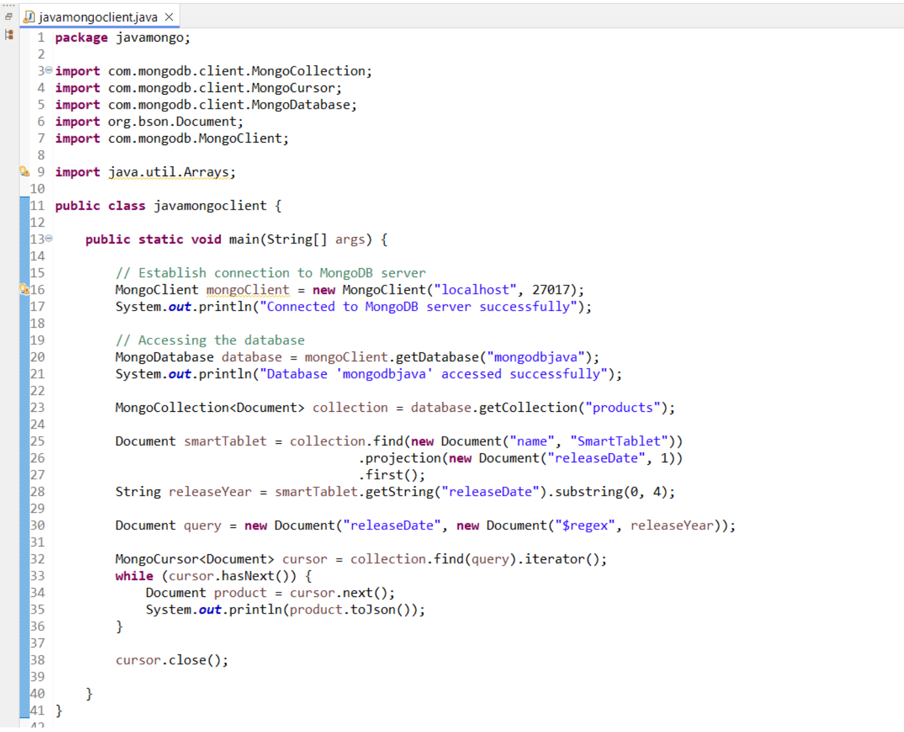
Assignment-2

Name: P CHARAN RAGAVENDRA Reg No: 21BCE1932

Questions:

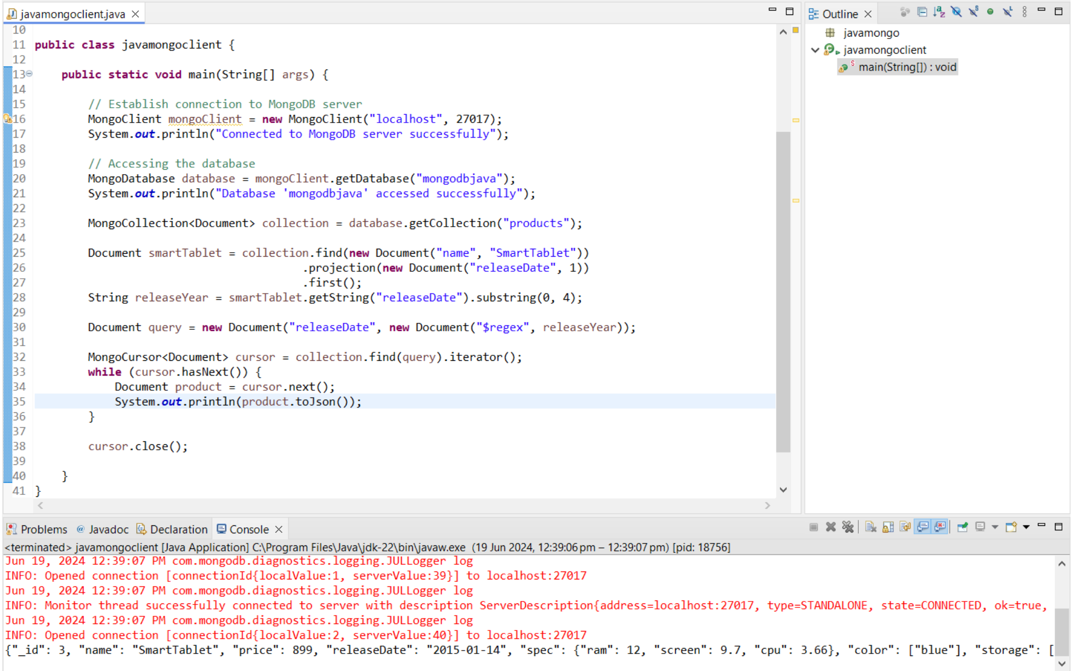
1.Find products released in the same year as "SmartTablet".

Code:



```java
package javamongo;

import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document smartTablet = collection.find(new Document("name", "SmartTablet"))
                                        .projection(new Document("releaseDate", 1))
                                        .first();
        String releaseYear = smartTablet.getString("releaseDate").substring(0, 4);

        Document query = new Document("releaseDate", new Document("$regex", releaseYear));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();

    }
}
```

Output:

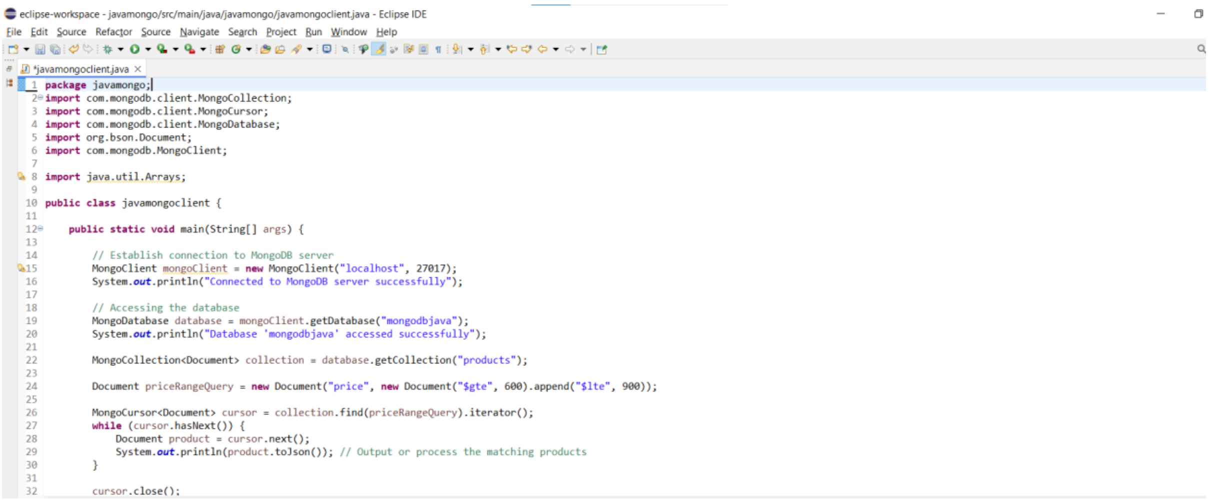


```java
public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document smartTablet = collection.find(new Document("name", "SmartTablet"))
                                        .projection(new Document("releaseDate", 1))
                                        .first();
        String releaseYear = smartTablet.getString("releaseDate").substring(0, 4);

        Document query = new Document("releaseDate", new Document("$regex", releaseYear));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();

    }
}
```

```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 12:39:06 pm – 12:39:07 pm) [pid: 18756]
Jun 19, 2024 12:39:07 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:39}] to localhost:27017
Jun 19, 2024 12:39:07 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true,
Jun 19, 2024 12:39:07 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:40}] to localhost:27017
{"_id": 3, "name": "SmartTablet", "price": 899, "releaseDate": "2015-01-14", "spec": {"ram": 12, "screen": 9.7, "cpu": 3.66}, "color": ["blue"], "storage": [
```

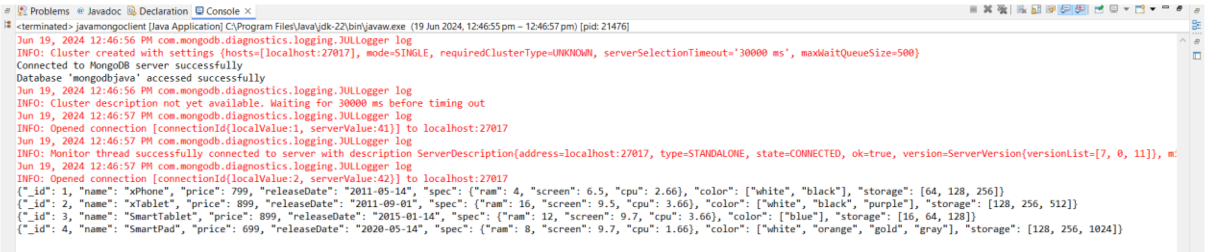
2.Find products where the price is between 600 and 900, inclusive

Code:


eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
*javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document priceRangeQuery = new Document("price", new Document("$gte", 600).append("$lte", 900));

        MongoCursor<Document> cursor = collection.find(priceRangeQuery).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
```

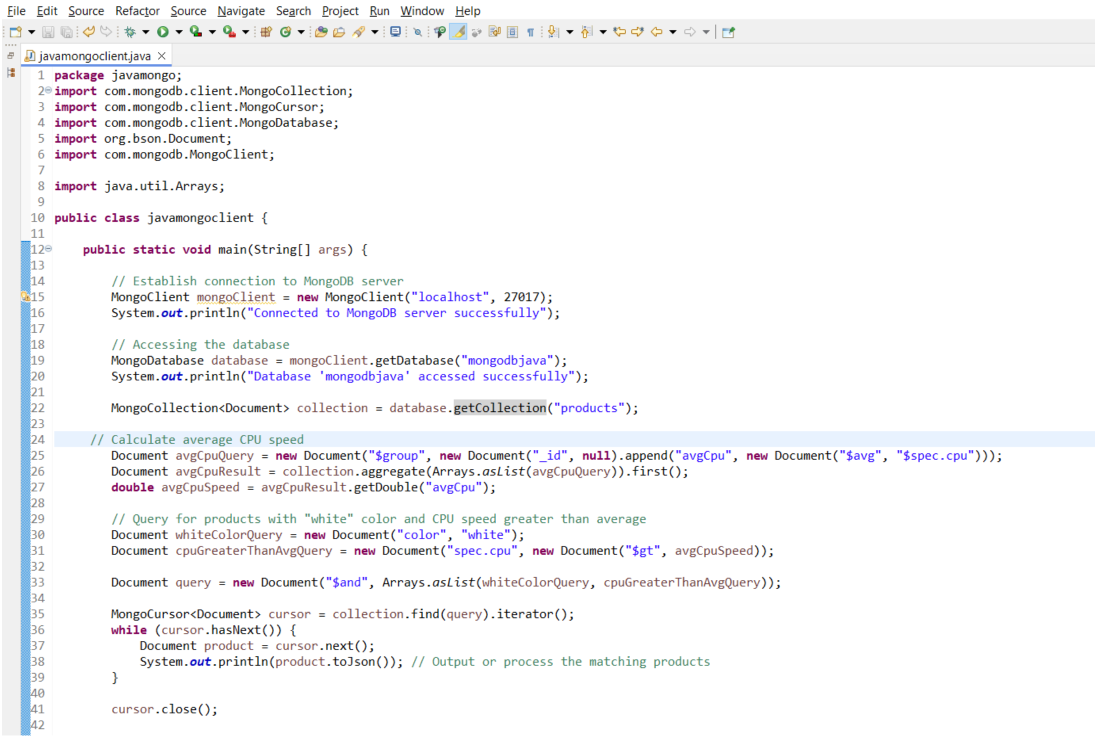
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (19 Jun 2024, 12:46:55 pm – 12:46:57 pm) [pid: 21476]
Jun 19, 2024 12:46:56 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 12:46:56 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 12:46:57 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:41}] to localhost:27017
Jun 19, 2024 12:46:57 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m
Jun 19, 2024 12:46:57 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:42}] to localhost:27017
{"_id": 1, "name": "xPhone", "price": 799, "releaseDate": "2011-05-14", "spec": {"ram": 4, "screen": 6.5, "cpu": 2.66}, "color": ["white", "black"], "storage": [64, 128, 256]}
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
{"_id": 3, "name": "SmartTablet", "price": 899, "releaseDate": "2015-01-14", "spec": {"ram": 12, "screen": 9.7, "cpu": 3.66}, "color": ["blue"], "storage": [16, 64, 128]}
{"_id": 4, "name": "SmartPad", "price": 699, "releaseDate": "2020-05-14", "spec": {"ram": 8, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256, 1024]}
```

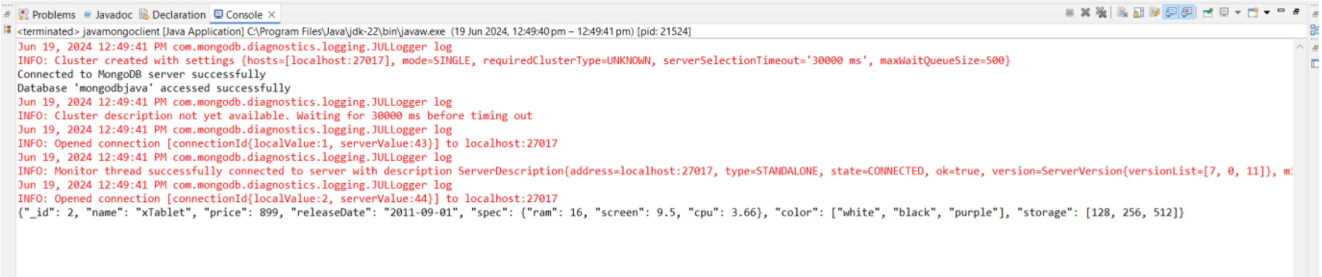
3.Find all products that have "white" as a color and a CPU speed greater than the average CPU speed of all products.

Code:


File Edit Source Refactor Source Navigate Search Project Run Window Help
javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

     // Calculate average CPU speed
        Document avgCpuQuery = new Document("$group", new Document("_id", null).append("avgCpu", new Document("$avg", "$spec.cpu")));
        Document avgCpuResult = collection.aggregate(Arrays.asList(avgCpuQuery)).first();
        double avgCpuSpeed = avgCpuResult.getDouble("avgCpu");

        // Query for products with "white" color and CPU speed greater than average
        Document whiteColorQuery = new Document("color", "white");
        Document cpuGreaterThanAvgQuery = new Document("spec.cpu", new Document("$gt", avgCpuSpeed));

        Document query = new Document("$and", Arrays.asList(whiteColorQuery, cpuGreaterThanAvgQuery));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
```

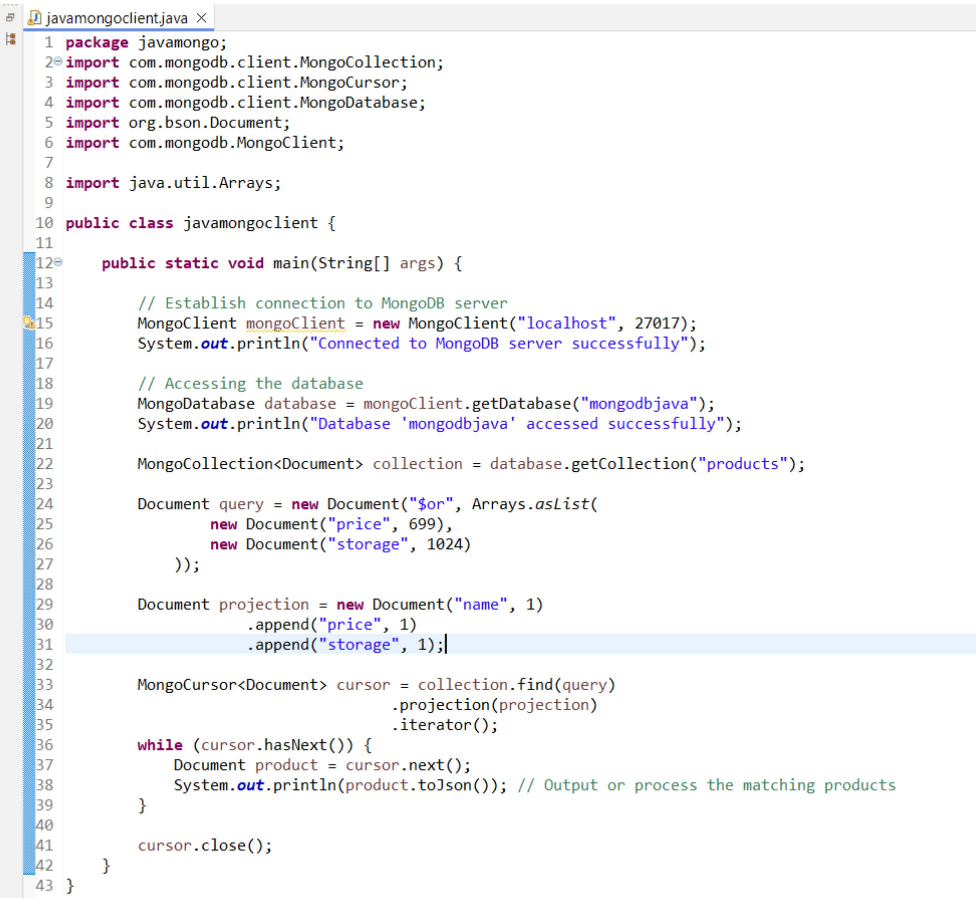
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 12:49:40 pm – 12:49:41 pm) [pid: 21524]
Jun 19, 2024 12:49:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 12:49:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 12:49:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:43}] to localhost:27017
Jun 19, 2024 12:49:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m
Jun 19, 2024 12:49:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:44}] to localhost:27017
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
```

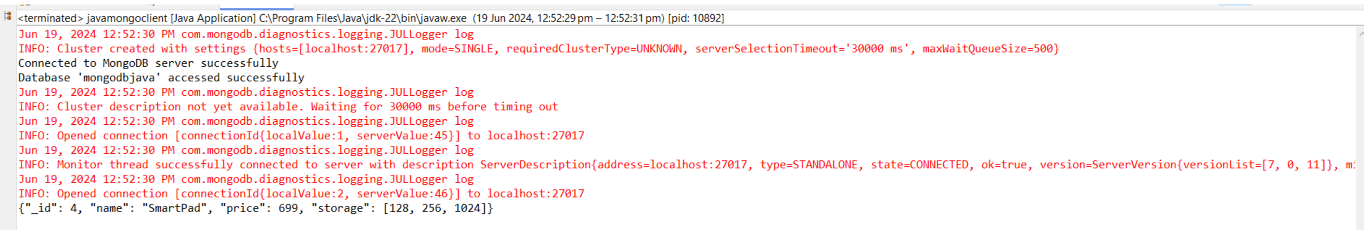
4.find name,price,storage if price is 699 or storage is 1024.

Code:



```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$or", Arrays.asList(
                new Document("price", 699),
                new Document("storage", 1024)
            ));

        Document projection = new Document("name", 1)
                            .append("price", 1)
                            .append("storage", 1);

        MongoCursor<Document> cursor = collection.find(query)
                                        .projection(projection)
                                        .iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
    }
}
```

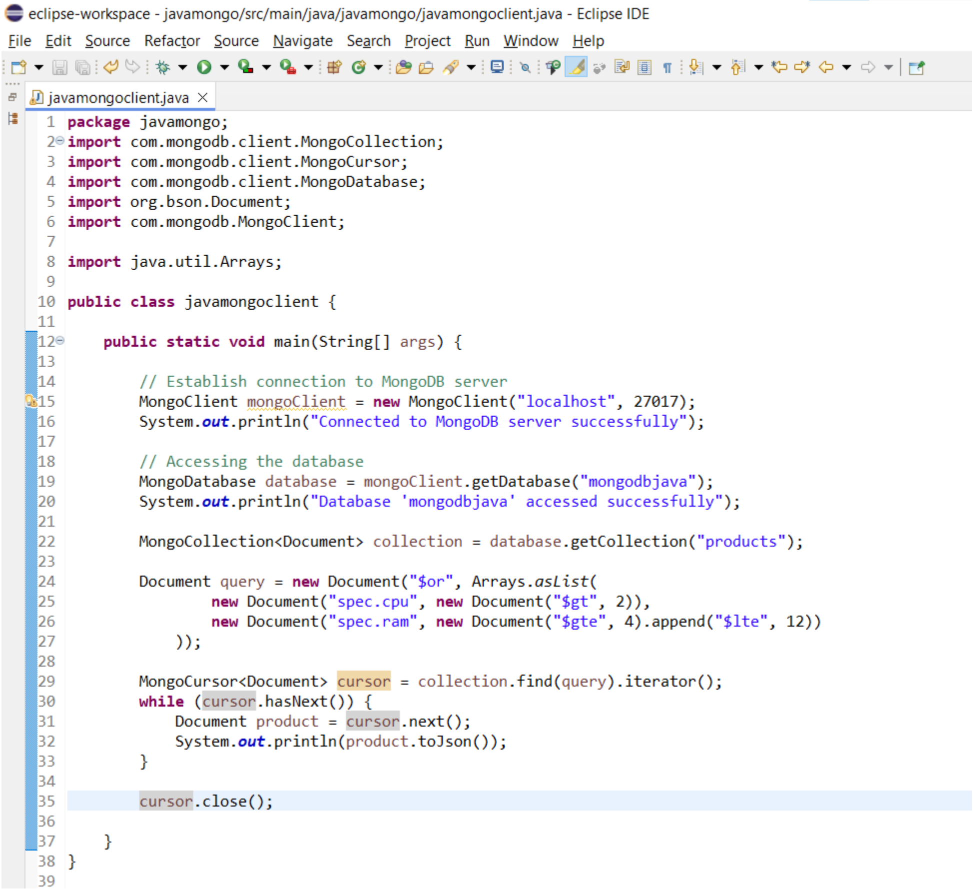
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 12:52:29 pm – 12:52:31 pm) [pid: 10892]
Jun 19, 2024 12:52:30 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 12:52:30 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 12:52:30 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:45}] to localhost:27017
Jun 19, 2024 12:52:30 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, m
Jun 19, 2024 12:52:30 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:46}] to localhost:27017
{"_id": 4, "name": "SmartPad", "price": 699, "storage": [128, 256, 1024]}
```

5.Find products with either cpu is greater than 2gb or ram is between 4 and 12.

Code:

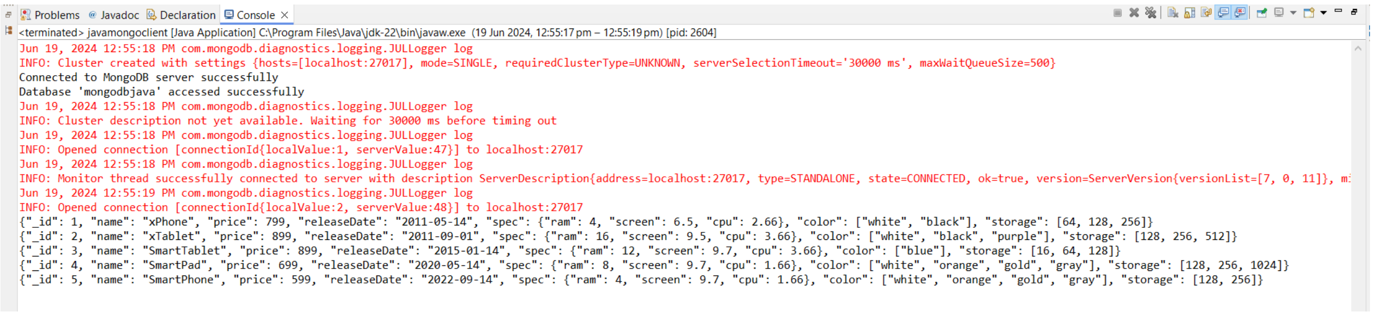


eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help
javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$or", Arrays.asList(
                new Document("spec.cpu", new Document("$gt", 2)),
                new Document("spec.ram", new Document("$gte", 4).append("$lte", 12))
        ));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();

    }
}
```

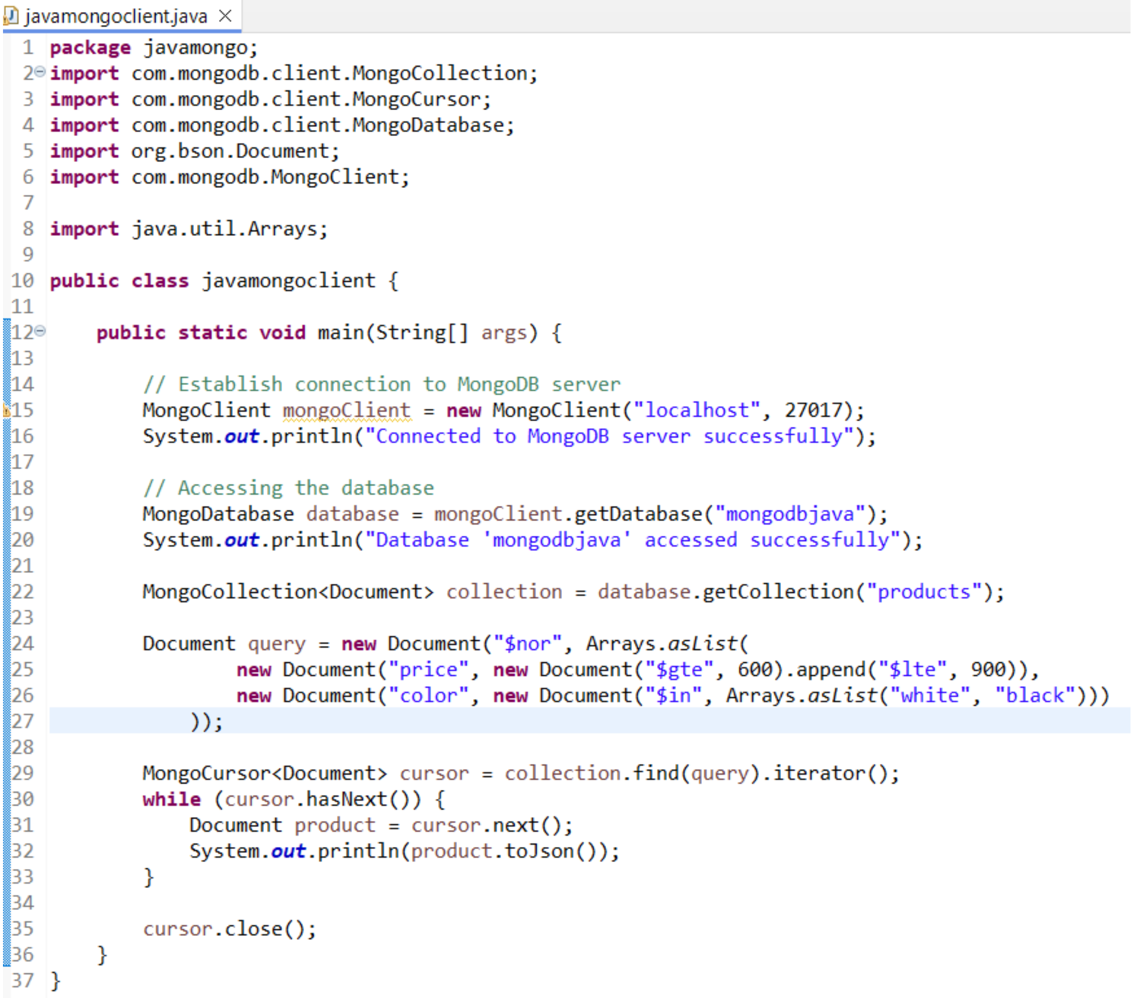
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (19 Jun 2024, 12:55:17 pm – 12:55:19 pm) [pid: 2604]
Jun 19, 2024 12:55:18 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 12:55:18 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 12:55:18 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:47}] to localhost:27017
Jun 19, 2024 12:55:18 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 12:55:19 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:48}] to localhost:27017
{"_id": 1, "name": "xPhone", "price": 799, "releaseDate": "2011-05-14", "spec": {"ram": 4, "screen": 6.5, "cpu": 2.66}, "color": ["white", "black"], "storage": [64, 128, 256]}
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
{"_id": 3, "name": "SmartTablet", "price": 899, "releaseDate": "2015-01-14", "spec": {"ram": 12, "screen": 9.7, "cpu": 3.66}, "color": ["blue"], "storage": [16, 64, 128]}
{"_id": 4, "name": "SmartPad", "price": 699, "releaseDate": "2020-05-14", "spec": {"ram": 8, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256, 1024]}
{"_id": 5, "name": "SmartPhone", "price": 599, "releaseDate": "2022-09-14", "spec": {"ram": 4, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256]}
```

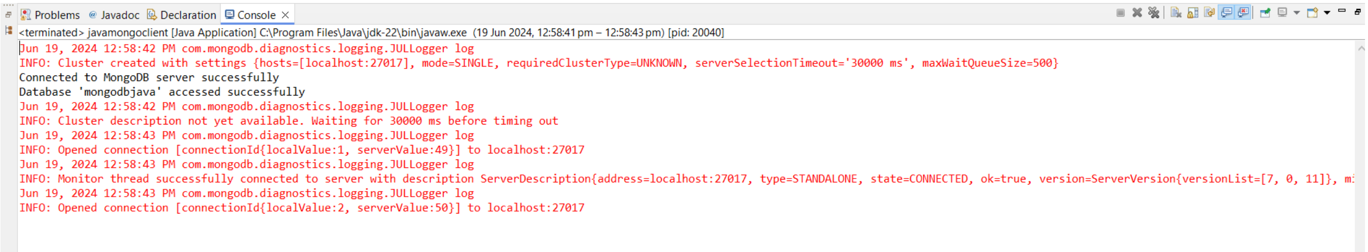
6. Find products with niether price is between 600 and 900 nor is it white or black.

Code:



```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$nor", Arrays.asList(
                new Document("price", new Document("$gte", 600).append("$lte", 900)),
                new Document("color", new Document("$in", Arrays.asList("white", "black")))
            ));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();
    }
}
```

Output:

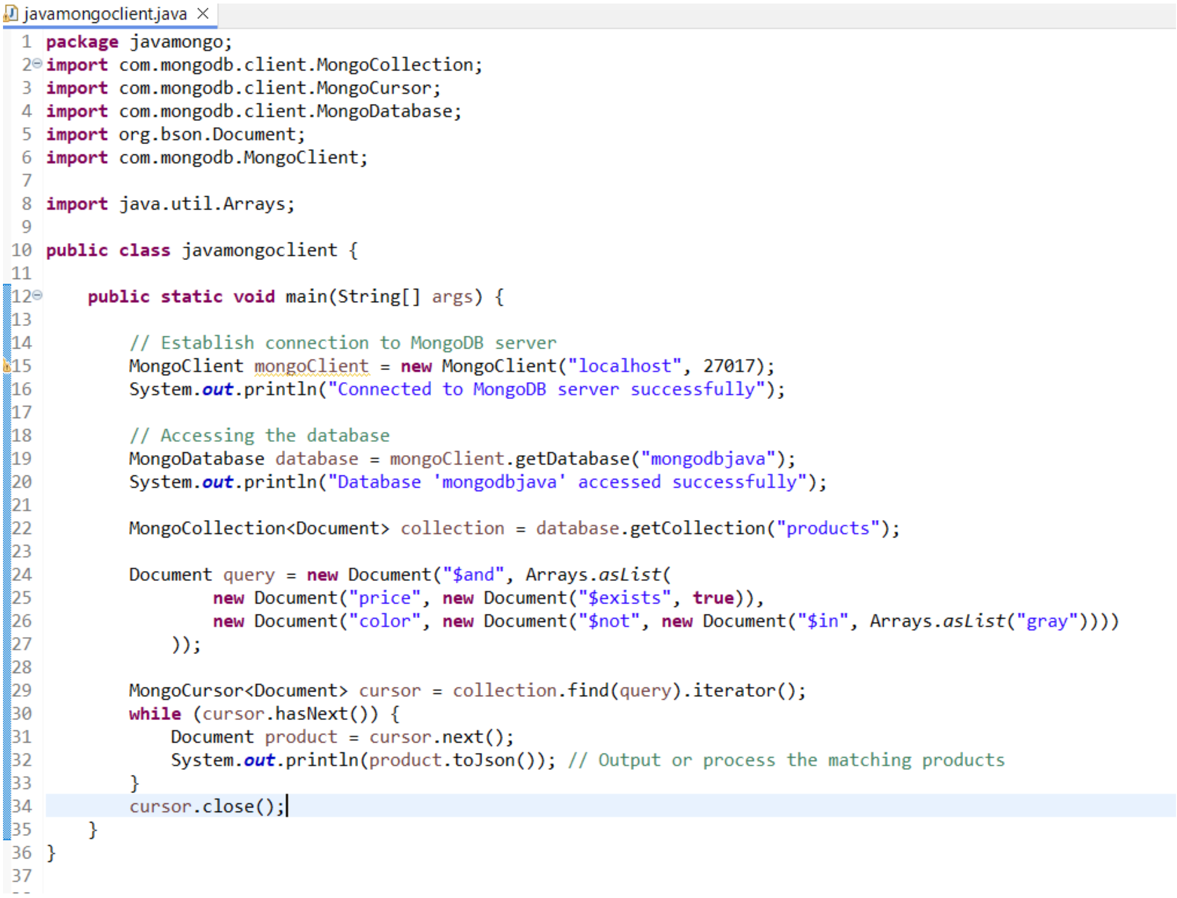


```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 12:58:41 pm – 12:58:43 pm) [pid: 20040]
Jun 19, 2024 12:58:42 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 12:58:42 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 12:58:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:49}] to localhost:27017
Jun 19, 2024 12:58:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 12:58:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:50}] to localhost:27017
```

(No information is displayed.)

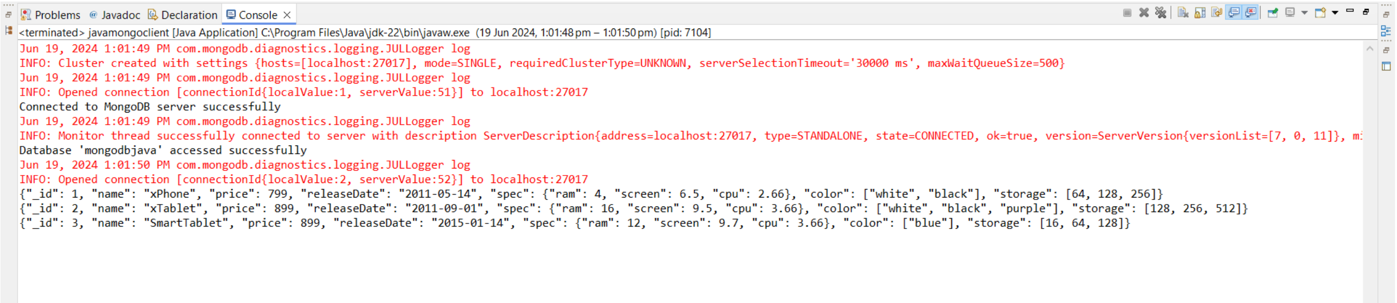
7. Find products where price exists and there is no gray in the colours.

Code:



```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$and", Arrays.asList(
                new Document("price", new Document("$exists", true)),
                new Document("color", new Document("$not", new Document("$in", Arrays.asList("gray"))))
            ));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }
        cursor.close();
    }
}
```

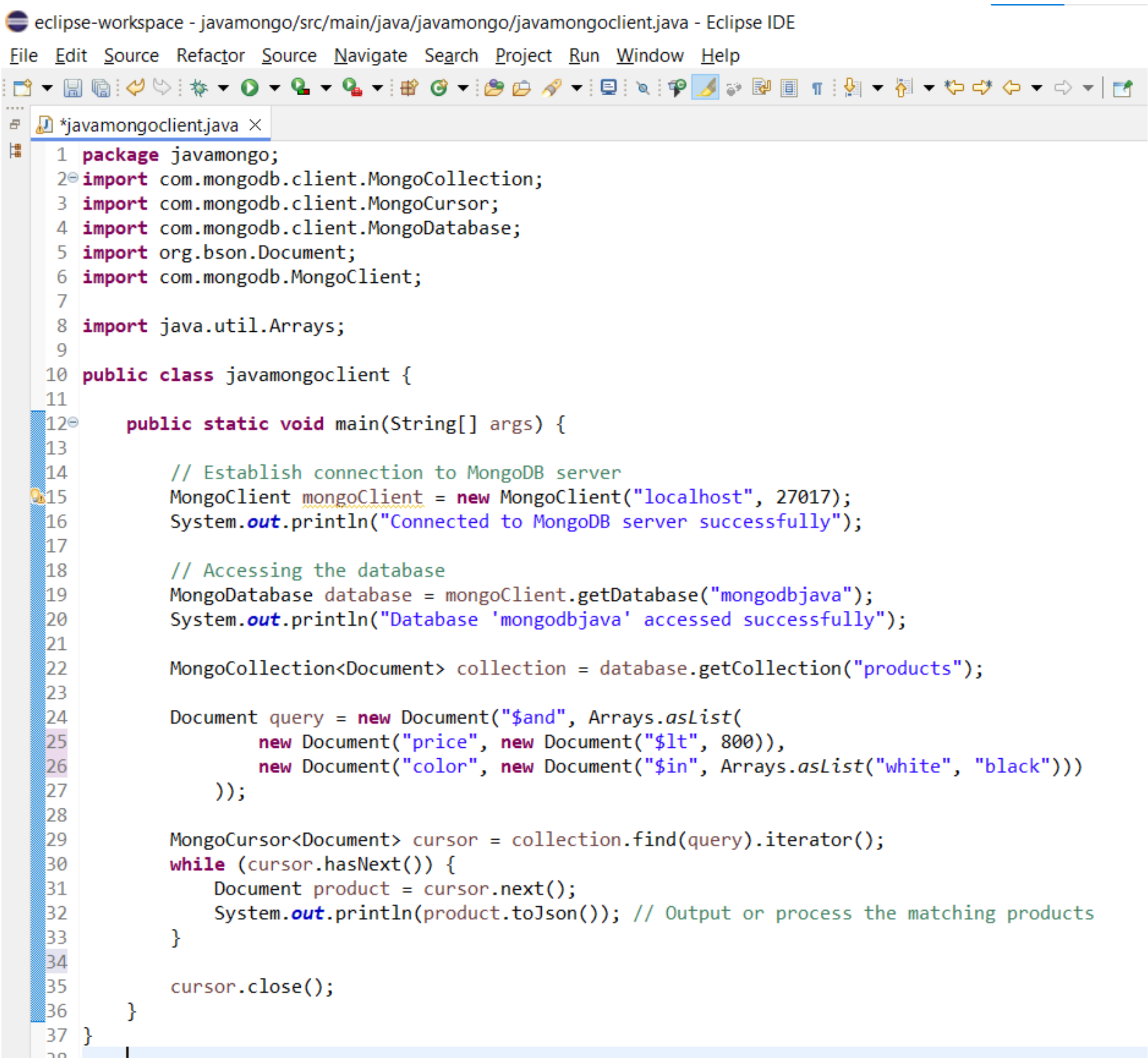
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 1:01:48 pm – 1:01:50 pm) [pid: 7104]
Jun 19, 2024 1:01:49 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Jun 19, 2024 1:01:49 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:51}] to localhost:27017
Connected to MongoDB server successfully
Jun 19, 2024 1:01:49 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:01:50 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:52}] to localhost:27017
{"_id": 1, "name": "xPhone", "price": 799, "releaseDate": "2011-05-14", "spec": {"ram": 4, "screen": 6.5, "cpu": 2.66}, "color": ["white", "black"], "storage": [64, 128, 256]}
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
{"_id": 3, "name": "SmartTablet", "price": 899, "releaseDate": "2015-01-14", "spec": {"ram": 12, "screen": 9.7, "cpu": 3.66}, "color": ["blue"], "storage": [16, 64, 128]}
```

8.Find products that have either "white" or "black" as a color option and are priced below 800.

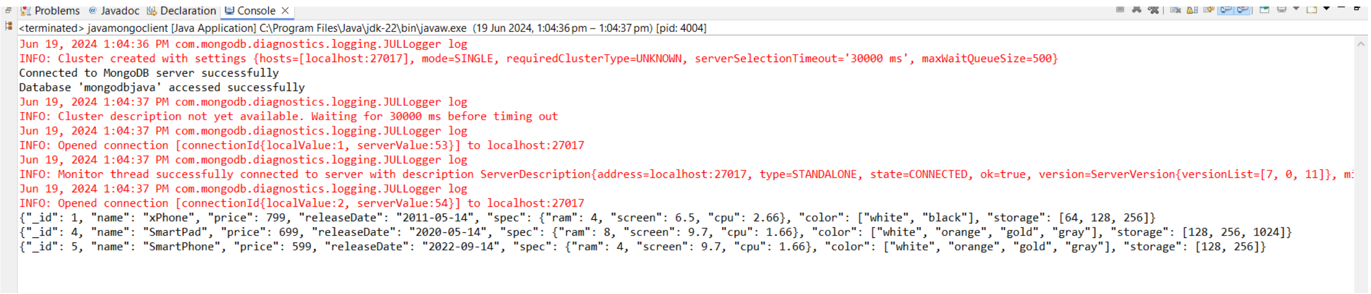
Code:



eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
*javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$and", Arrays.asList(
                new Document("price", new Document("$lt", 800)),
                new Document("color", new Document("$in", Arrays.asList("white", "black")))
            ));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
    }
}
```

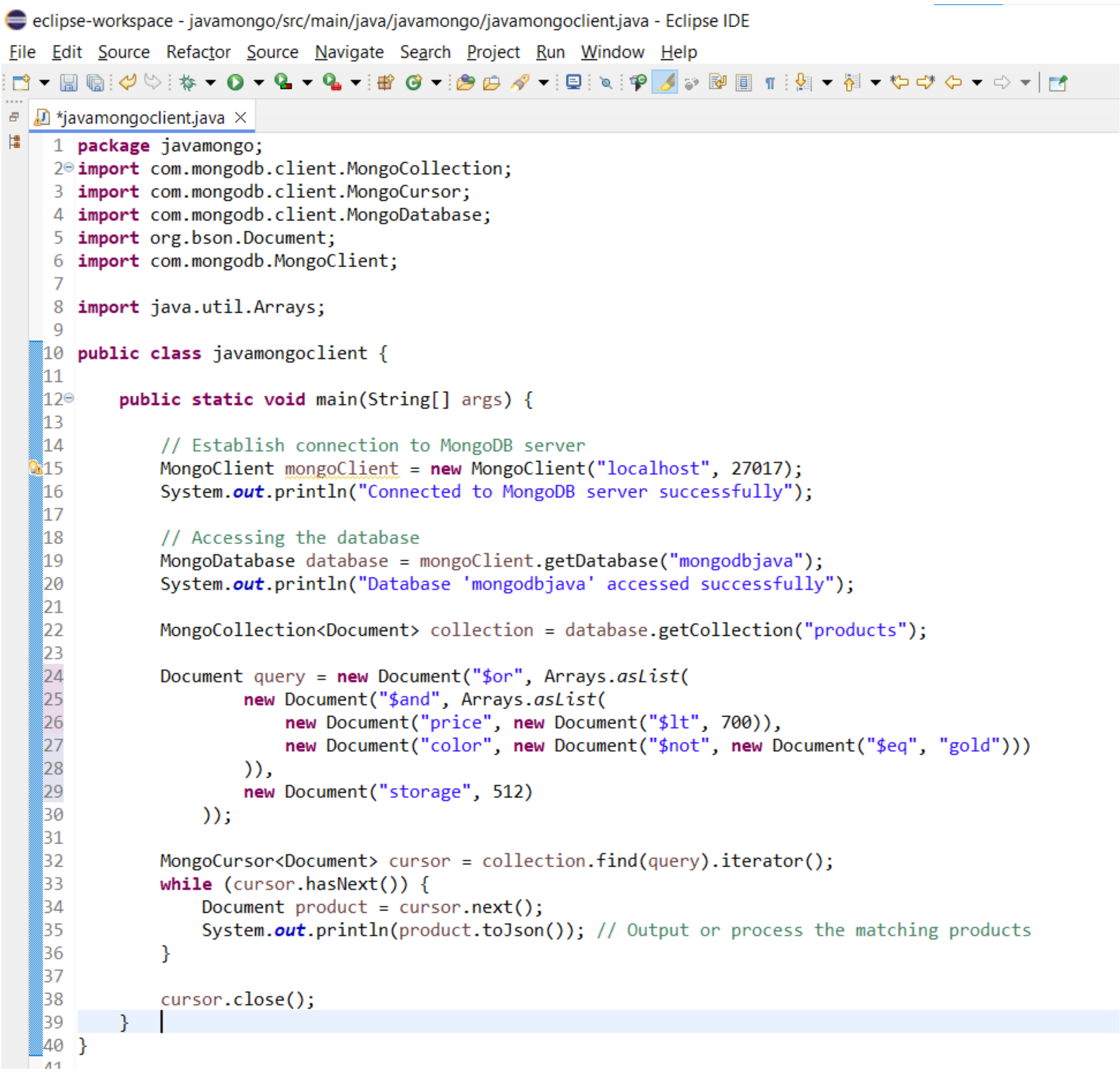
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 1:04:36 pm – 1:04:37 pm) [pid: 4004]
Jun 19, 2024 1:04:36 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:04:37 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 1:04:37 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:53}] to localhost:27017
Jun 19, 2024 1:04:37 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 1:04:37 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:54}] to localhost:27017
{"_id": 1, "name": "xPhone", "price": 799, "releaseDate": "2011-05-14", "spec": {"ram": 4, "screen": 6.5, "cpu": 2.66}, "color": ["white", "black"], "storage": [64, 128, 256]}
{"_id": 4, "name": "SmartPad", "price": 699, "releaseDate": "2020-05-14", "spec": {"ram": 8, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256, 1024]}
{"_id": 5, "name": "SmartPhone", "price": 599, "releaseDate": "2022-09-14", "spec": {"ram": 4, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256]}
```

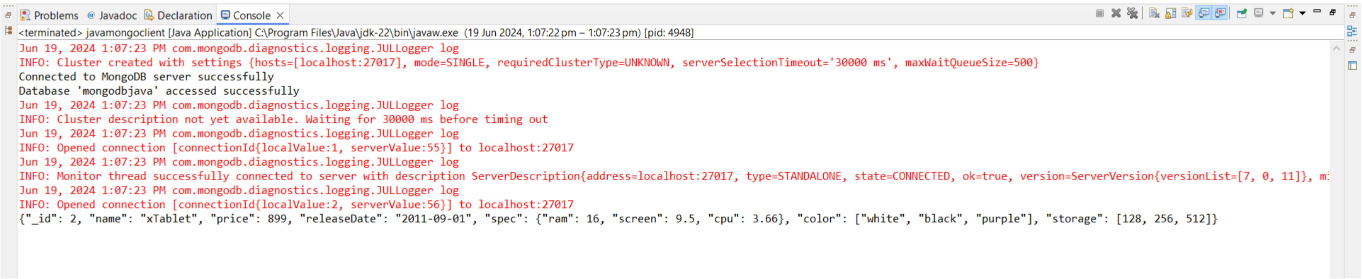
9.Find products that do not have "gold" as a color and are priced below 700 or have a storage option of 512GB.

Code:


eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
*javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("$or", Arrays.asList(
                new Document("$and", Arrays.asList(
                    new Document("price", new Document("$lt", 700)),
                    new Document("color", new Document("$not", new Document("$eq", "gold")))
                )),
                new Document("storage", 512)
            ));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
    }
}
```

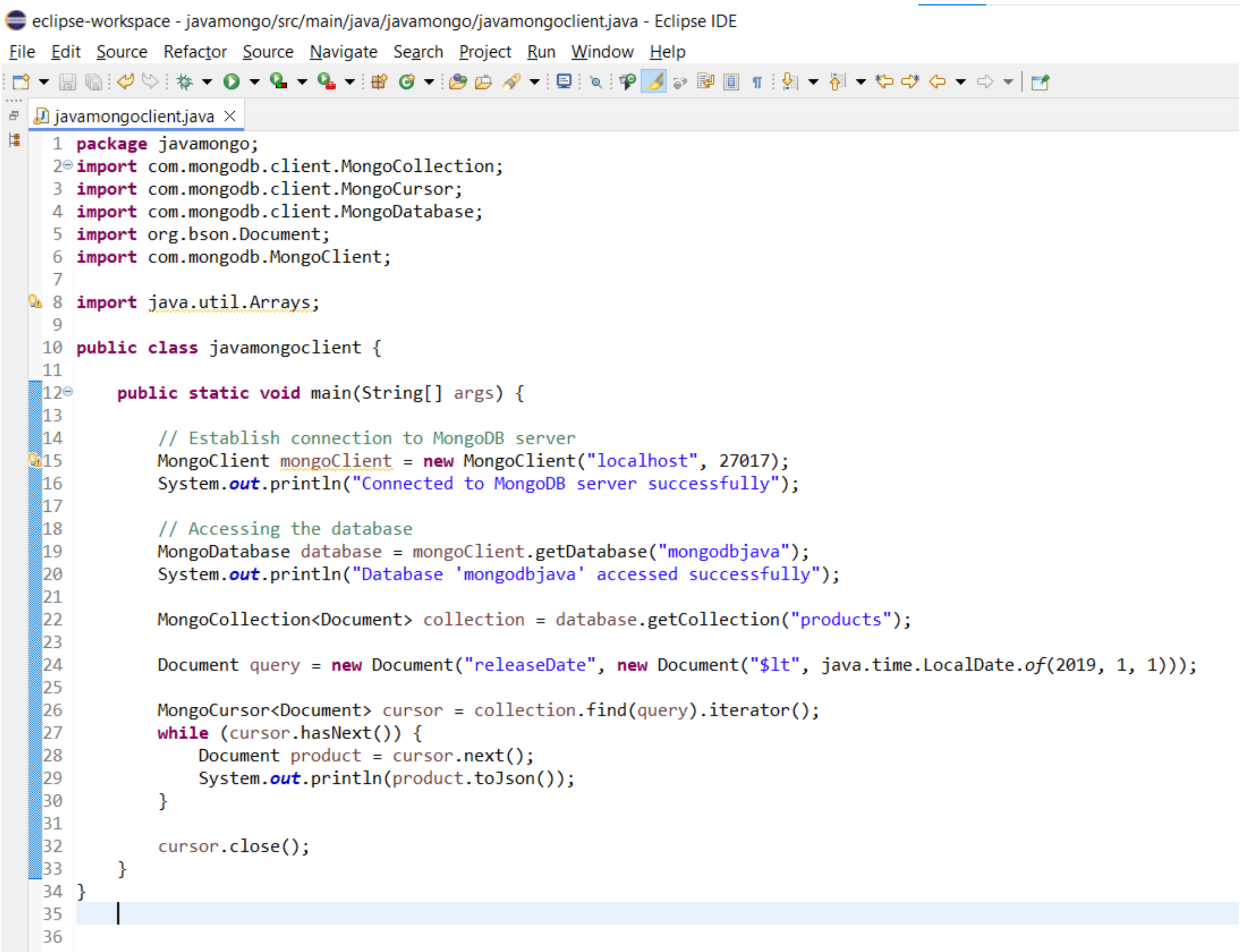
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 1:07:22 pm – 1:07:23 pm) [pid: 4948]
Jun 19, 2024 1:07:23 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:07:23 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 1:07:23 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:55}] to localhost:27017
Jun 19, 2024 1:07:23 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 1:07:23 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:56}] to localhost:27017
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
```

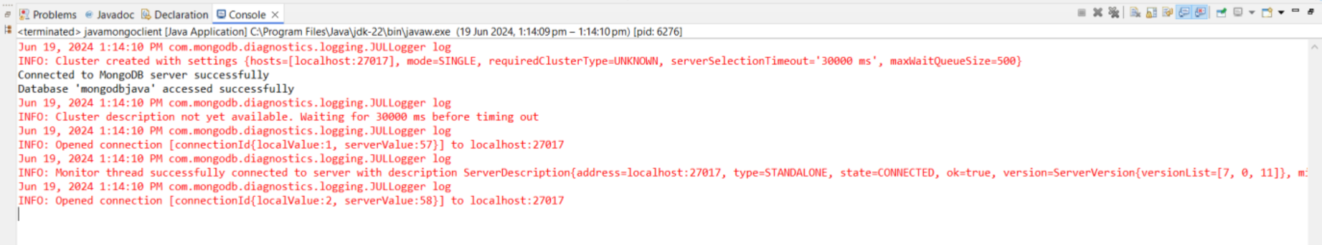
10. Find products released before 2019.

Code:


eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("releaseDate", new Document("$lt", java.time.LocalDate.of(2019, 1, 1)));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();
    }
}
```

Output:

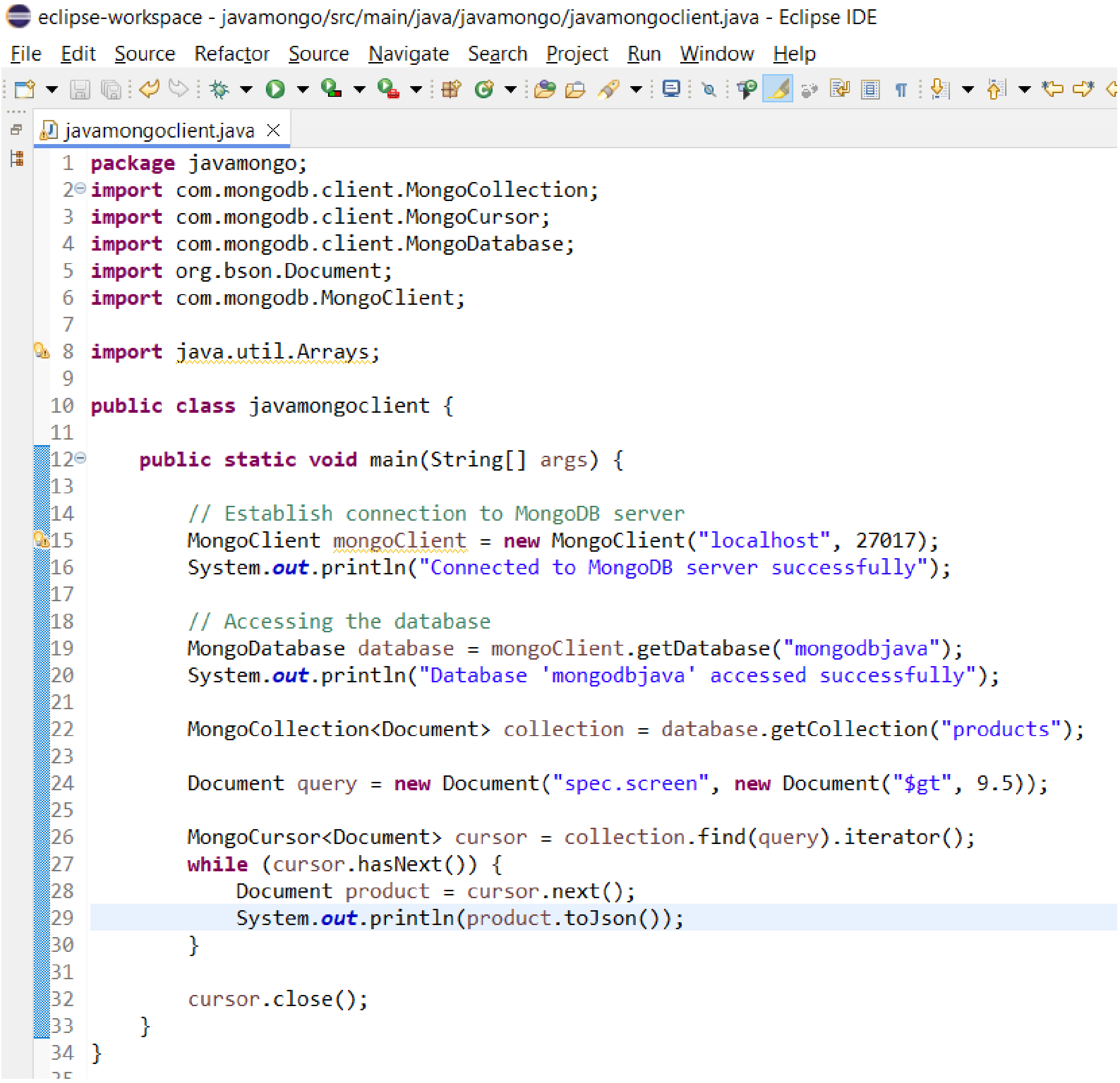


```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 1:14:09 pm – 1:14:10 pm) [pid: 6276]
Jun 19, 2024 1:14:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:14:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 1:14:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:57}] to localhost:27017
Jun 19, 2024 1:14:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 1:14:10 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:58}] to localhost:27017
```

(No information displayed)

11. List products with a screen size greater than 9.5 inches.

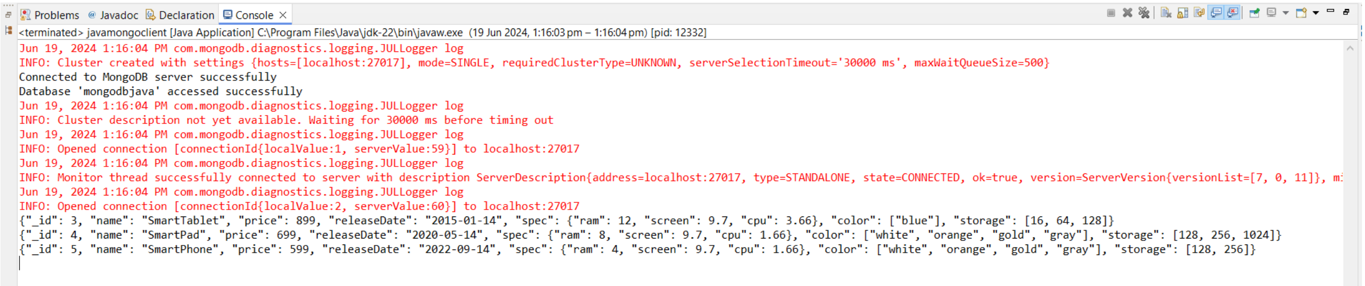
Code:



eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("spec.screen", new Document("$gt", 9.5));

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson());
        }

        cursor.close();
    }
}
```

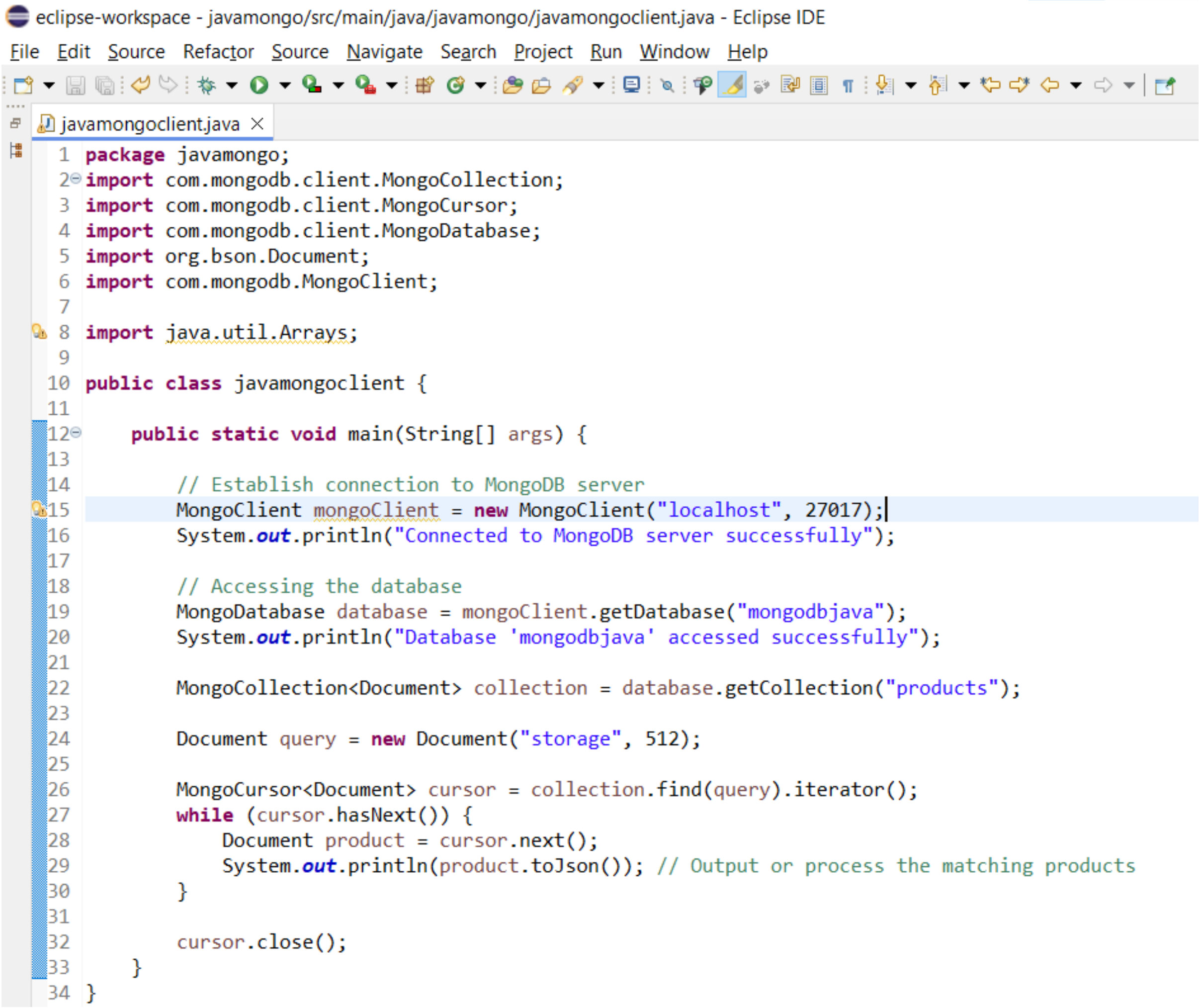
Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (19 Jun 2024, 1:16:03 pm – 1:16:04 pm) [pid: 12332]
Jun 19, 2024 1:16:04 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:16:04 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 1:16:04 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:59}] to localhost:27017
Jun 19, 2024 1:16:04 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 1:16:04 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:60}] to localhost:27017
{"_id": 3, "name": "SmartTablet", "price": 899, "releaseDate": "2015-01-14", "spec": {"ram": 12, "screen": 9.7, "cpu": 3.66}, "color": ["blue"], "storage": [16, 64, 128]}
{"_id": 4, "name": "SmartPad", "price": 699, "releaseDate": "2020-05-14", "spec": {"ram": 8, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256, 1024]}
{"_id": 5, "name": "SmartPhone", "price": 599, "releaseDate": "2022-09-14", "spec": {"ram": 4, "screen": 9.7, "cpu": 1.66}, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256]}
```

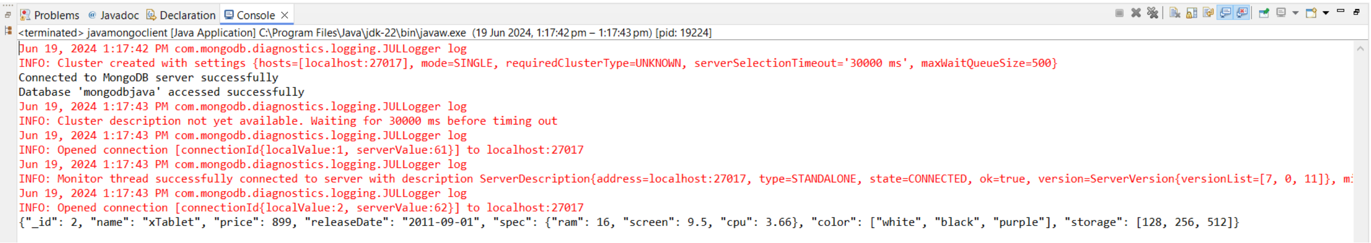
12. Retrieve products with storage options including 512 GB.

Code:


eclipse-workspace - javamongo/src/main/java/javamongo/javamongoclient.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
javamongoclient.java

```java
package javamongo;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import com.mongodb.MongoClient;

import java.util.Arrays;

public class javamongoclient {

    public static void main(String[] args) {

        // Establish connection to MongoDB server
        MongoClient mongoClient = new MongoClient("localhost", 27017);
        System.out.println("Connected to MongoDB server successfully");

        // Accessing the database
        MongoDatabase database = mongoClient.getDatabase("mongodbjava");
        System.out.println("Database 'mongodbjava' accessed successfully");

        MongoCollection<Document> collection = database.getCollection("products");

        Document query = new Document("storage", 512);

        MongoCursor<Document> cursor = collection.find(query).iterator();
        while (cursor.hasNext()) {
            Document product = cursor.next();
            System.out.println(product.toJson()); // Output or process the matching products
        }

        cursor.close();
    }
}
```

Output:



```
<terminated> javamongoclient [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (19 Jun 2024, 1:17:42 pm – 1:17:43 pm) [pid: 19224]
Jun 19, 2024 1:17:42 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to MongoDB server successfully
Database 'mongodbjava' accessed successfully
Jun 19, 2024 1:17:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
Jun 19, 2024 1:17:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:61}] to localhost:27017
Jun 19, 2024 1:17:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[7, 0, 11]}, mi
Jun 19, 2024 1:17:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:62}] to localhost:27017
{"_id": 2, "name": "xTablet", "price": 899, "releaseDate": "2011-09-01", "spec": {"ram": 16, "screen": 9.5, "cpu": 3.66}, "color": ["white", "black", "purple"], "storage": [128, 256, 512]}
```