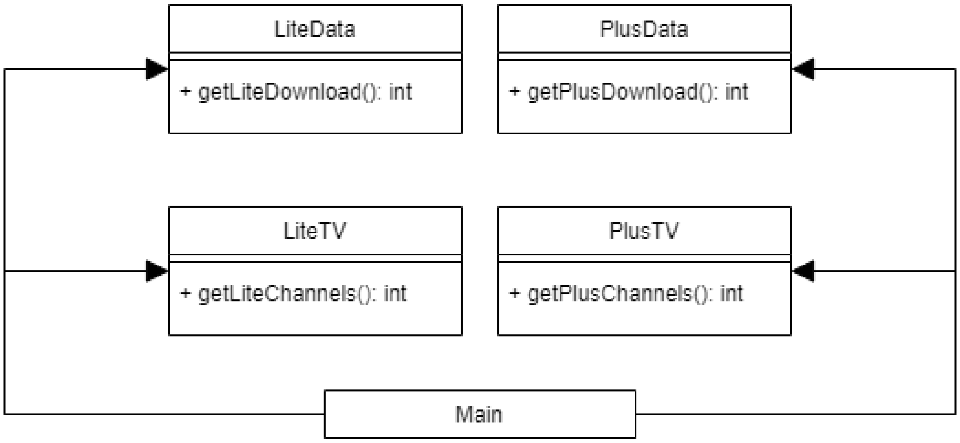
Answer the questions below based on the following scenario:

A cable service provider has an application that manages the packages it offers to customers. A package consists of a Data and a TV bundle. The class diagram below outlines the functionality for representing the various bundles.



The Main class calls the bundle classes to create packages and have them print details about the package.

```
class Main {
 public static void main(String[] args) {
  String[] packages = {"Lite", "Plus", "Lite", "Lite"};
  for (String p: packages){
    if ( p == "Lite"){ //create a lite package
      LiteData ld = new LiteData();
      LiteTV lt = new LiteTV();
      System.out.println(
       p+" package has "+
       ld.getLiteDownload()+" download and "+
       lt.getLiteChannels()+" channels"
      );
    }else if (p == "Plus"){ //create a plus package
      PlusData pd = new PlusData();
      PlusTV pt = new PlusTV();
      System.out.println(
       p+" package has "+
       pd.getPlusDownload()+" download and "+
       pt.getPlusChannels()+ " channels"
      );
    }
  }
 }
}
```

The code above produces the following output:

```
Lite package has 100 download and 32 channels
Plus package has 250 download and 68 channels
Lite package has 100 download and 32 channels
Lite package has 100 download and 32 channels
```

Questions

1. By analyzing the UML diagram, identify any problem areas with the software design.
2. By analyzing the code, describe an issue with the implementation citing any solid violations
3. Identify a creational design pattern (Singleton, Factory Method, Abstract Factory, Builder) that addresses an issue in the current design and explain how.
4. Propose a redesign in the form of a UML class diagram that illustrates how your identified pattern improves design.
5. Write code for the refactored version of main that uses your selected pattern and highlight the benefit of its application with relevant code comments.