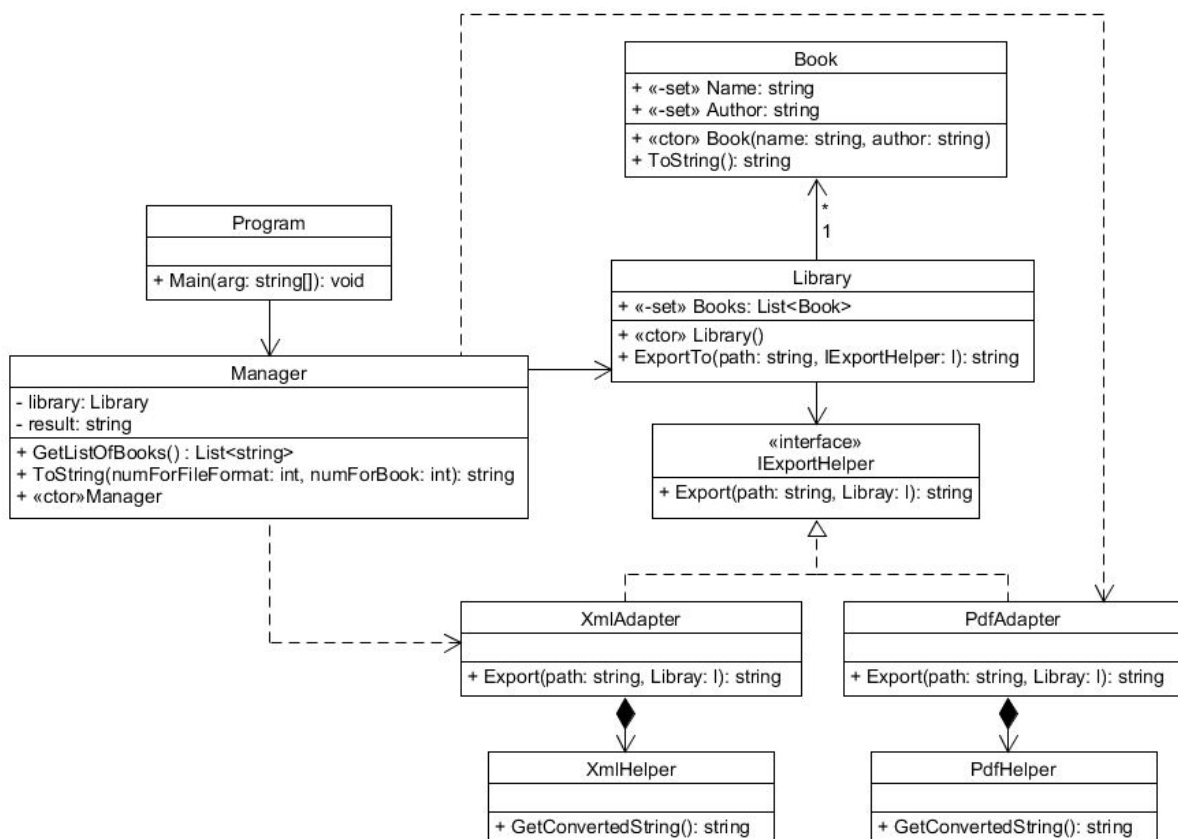


# Adapter Pattern

Mrunal Patil - 3101851  
Emil Karamihov - 2992884

**Application Description:** The application is to convert Books to different text formats. Eg:- Books are converted to xml and/or pdf. The Console has an object of the Manager, Manager has information about all the Adapters and the Library. Adapters are the links to combine file helpers with the library. The adapter implements the interface IExportHelper. The interface has a method export with pathname and library as parameters where path is the path of the file and library is the library used to convert the file type.



## Maintainability, Reusability, Extensibility.

Maintainability is one of the strong suits of this pattern, because of the reason that the code is isolated. If more functionality is being added it is as simple as adding the required methods to the adaptee, so the adapter can use them. If a conversion system must be removed, it is as simple as just removing the reference of the adapter from the manager. Client must never be changed.

Reusability this pattern is very popular for reusability since some of the adaptees can be also use other adaptees to make the solution work, but this can be a drawback. For the main reason if one of the converters is removed some of the adaptees might be dependent.

Extensibility may be a problem since for every adaptee there must be created an adapter to connect the it to the application. So intotal two classes for one functionality. If more functionality is created twice the classes must be made.

**Pros**

1. Hides from the client code unnecessary implementation details of interface & data conversion.
2. Only the manager has an instance of the adapters, so if a new adaptee is added, there must be an instance in the manager
3. Client must not be aware of all the different interfaces in use
4. Adapter can use multiple adaptees, without exposing the complexity to the client
5. When adaptee changes only the adapter needs to be updated
6. You can easily change the library used for conversion of file in the adapter without affecting any other method or class

**Cons**

1. Increases overall code complexity by creating additional classes. Communication overhead