C# & OOP

Activity 1. Arrays and writing to files

Create an array which will hold the 12 most highly paid actors in Hollywood.

Place the following names in the array elements:

1. Tom Cruise
2. Brad Pitt
3. Julia Roberts
4. Angelina Jolie
5. Jennifer Aniston
6. Tom Hanks
7. Johnny Depp
8. Cameron Diaz
9. Adam Sandler
10. Will Smith
11. Ben Stiller
12. Reese Witherspoon

Display the actors name one per line.

Sort the array and then display the actors names in sorted order (sorted by first name)

Save the list of actors to a text file named actors.txt, one per line.

Activity 2. Reading from files

Read the list of actors from the file actors.txt display them to a web page one per line.

Activity 3. Create a class

Create a class called celebrity:

Celebrity
- FirstName : string
- LastName : string
- Country : string
- Income : decimal
- Age : int
+ ProfileDescription()

Business rules to implement:

- FirstName, LastName and country are strings between 1 and 50 characters long with no leading or trailing white space
- Income is a decimal
- Age is an integer between 0 and 130.
- ProfileDescription returns a string which says *FirstName LastName Age* years old was born in *Country* and has earned *Income*.

Activity 4.   Inheritance

Now create 2 more classes named Actor and Musician. These 2 classes will inherit from the celebrity class.

Musician
-     Hits : ArrayList -   Genre :string
+ ListHits() + AddHit(string)

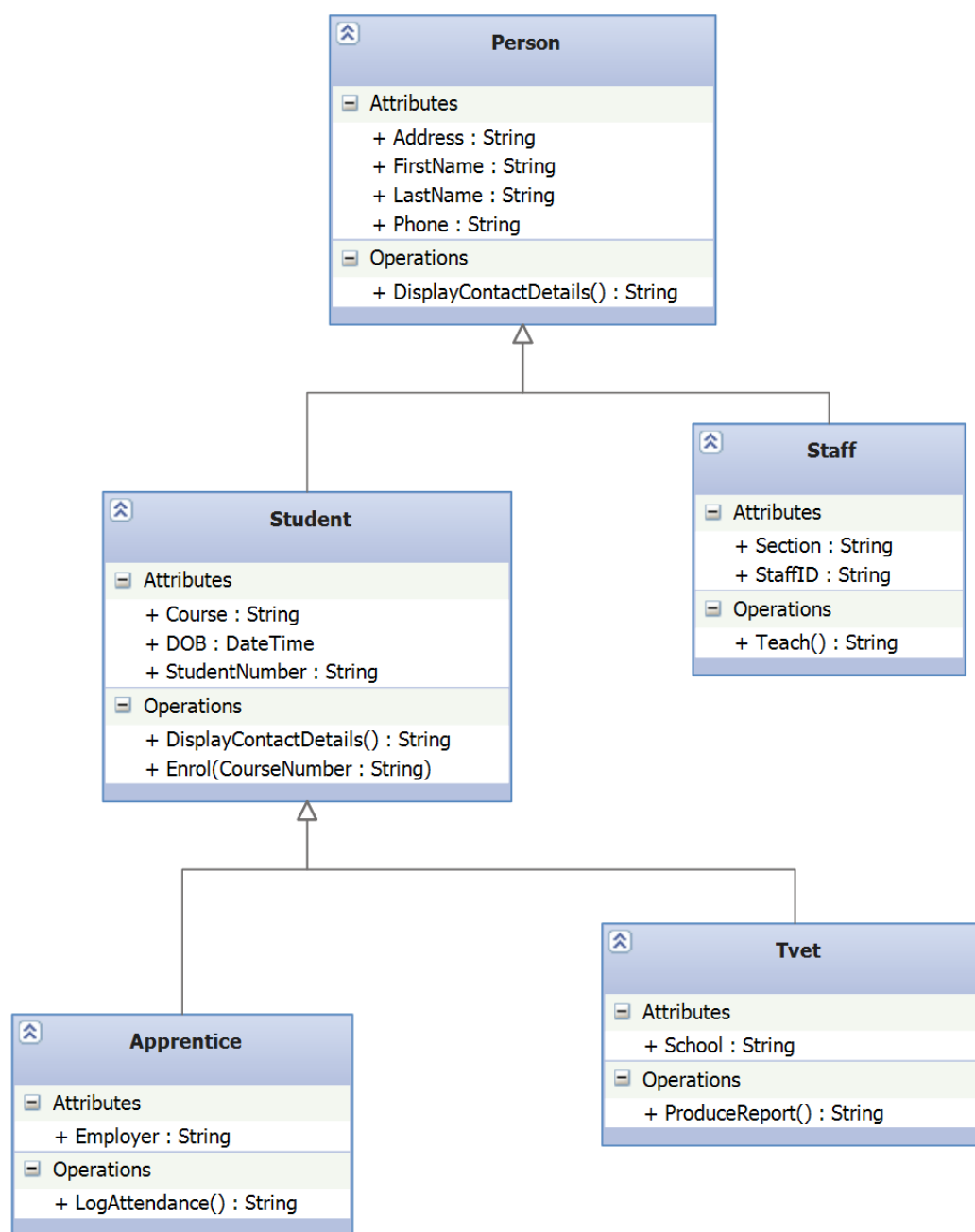| Actor |
| --- |
| -   LastJob : string
-     MostFamousFor : string
-     Email : string |
| + BookInterview(date, location) |

Business rules to implement

- Hits contains an arrayList of hits for this artist. Each hit is a string between 1 and 50 characters long with no leading or trailing white space. Hits is an array list which is private and can only be accessed by the methods listHits and addHit.
- Genre – string between 1 and 50 characters long with no leading or trailing white space
- ListHits returns a string containing all the hits separated by a comma.
- AddHit – adds a hit to the Hits arrayList. Each hit is a string between 1 and 50 characters long with no leading or trailing white space.
- LastJob is a string between 1 and 50 characters long with no leading or trailing white space
- MostFamousFor is a string between 1 and 50 characters long with no leading or trailing white space
- Email is a string between 1 and 250 characters long with no leading or trailing white space
- BookInterview needs a data and location as input and will send the celebrity an email to them inviting them to an interview

Activity 5.   Inheritance

Create the code to match the following class diagram:

Implement the following rules:

Person – FirstName and LastName are strings between 1 and 50 characters long with no leading or trailing blank spaces. Address is between 1 and 250 characters long with no leading or trailing blank spaces. Phone is between 1 and 25 characters long with no leading or trailing blank spaces. DisplayContactDetails returns a string listing the person's contact details.

Student – StudentNumber is exactly 10 characters long with no leading or trailing blank spaces. DOB is a date, Course is a 6 digit string. Enrol takes as input a course number and stores it in the Course property. The student class needs to override DisplayContactDetails so that the student number and course number are displayed.

C : \ a r i a n e \ T A F E \ c l a s s e s \ w e b d i p l o m a \ c l a s s m a t e r i a l \ O O P \ n e w \ c
s h a r p a n d o o p . d o c x

Page 3 of 4
Created: 3 May. 12

Staff – StaffID is exactly 10 characters long with no leading or trailing blank spaces. Section strings between 1 and 50 characters long with no leading or trailing blank spaces. Teach returns the string "first name, last name is now teaching"

Apprentice – Employer is between 1 and 250 characters long with no leading or trailing blank spaces. LogAttendance returns a string "Attendance logged"

Tvet – School is between 1 and 100 characters long with no leading or trailing blank spaces. ProduceReport returns a string listing the student name and their school.