

Abstract Class and Interfaces

1. Create an Interface **Instrument** which has play as a method. Create three classes from Instrument which is Piano, Flute, and Guitar. Override the play

Method inside all three classes, printing a message

“Piano is playing pee peee peee ” for Piano class

“Flute is playing toot toot toot toot” for Flute class

“Guitar is playing tin tin tin tin ” for Guitar class

Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method.

2. Create an Interface **Game** which is having start, play, and stop methods. Create classes Car, Bike, Bicycle by implementing Game interface. Create an array of 10 Games. Generate 10 random numbers in the range 1 to 3. If random number is 1 – add Car, 2-add Bike, 3- add Bicycle objects to the array. Process the elements from the array and display details of each object.

Hint: to generate the random number use Math class api.

3. public interface RemoteControl {

boolean powerOnOff(); // Returns new state, on = true

int volumeUp(int increment); // Returns new volume level

int volumeDown(int decrement); // Returns new volume level

void mute(); // Mutes sound output

} Create DVD, TV remote classs implementing the above interface. By using RemoteControl interface reference , check the dynamic polymorphism.

Hint:

For DVD, TV class, take MAX_SOUND, MIN_SOUND.

4. Create interface Currency with getCurrency method. Create classes India, USA, UK by implementing the Currency interface.

Create class CurrencyContainer, Currency getCurrency(String country) (Method return type Currency).

getCurrency(string country) Based on the country name it must return particular country object. If country is not valid it should throw IllegalArgumentException.

Abstract Class and Interfaces

Create class CurrencyClient with main method and prompt user to enter country name. By using country name, get Currency object by invoking the CurrencyContainer getCurrency() without creating object of CurrencyContainer and display the currency name by invoking getCurrency() Currency interface.

5. Create Interface MyIterator with boolean hasNext(),String next().

Create class UserList, String name[], UserList(), MyIterator getIterator(). When UserList object is created it must initialize name[] with 10 names. Don't allow to access name[] outside the class.

Create class UserClient with main method. Get Instance of userList and display all the users.

Hint:

Use MyIterator interface to process the elements from the name[].

Try to give implementation of MyIterator Interface in UserList class.