1. Create an Entity class 'Trainee', with traineeId, traineeName, contactNo, email, gender, age as fields. Create a class 'Batch', with batchCode, startdate, enddate and Trainee[] as fields. Create the following overloaded methods in the 'Batch' class public Trainee getTrainee(int traineeId) thows TraineeNotFoundException public Trainee[] getTrainees(String gender)

**package** com.css.corejava;

**import** java.util.Arrays;

**public** **class** Batch {

**private** String batchCode;

**private** String startDate;

**private** String endDate;

**private** Trainee[] trainees;

@Override

**public** String toString() {

**return** "Batch [batchCode=" + batchCode + ", startDate=" + startDate + ", endDate=" + endDate + ", trainees="

+ Arrays.*toString*(trainees) + "]";

}

**public** Batch(String batchCode, String startDate, String endDate, Trainee[] trainees) {

**super**();

**this**.batchCode = batchCode;

**this**.startDate = startDate;

**this**.endDate = endDate;

**this**.trainees = trainees;

}

**public** Batch() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** String getBatchCode() {

**return** batchCode;

}

**public** **void** setBatchCode(String batchCode) {

**this**.batchCode = batchCode;

}

**public** String getStartDate() {

**return** startDate;

}

**public** **void** setStartDate(String startDate) {

**this**.startDate = startDate;

}

**public** String getEndDate() {

**return** endDate;

}

**public** **void** setEndDate(String endDate) {

**this**.endDate = endDate;

}

**public** Trainee[] getTrainees() {

**return** trainees;

}

**public** **void** setTrainees(Trainee[] trainees) {

**this**.trainees = trainees;

}

**public** Trainee[] getTrainees(String gender) {

Trainee[] foundTrainees=**new** Trainee[0];

**int** i=0;

**for** (Trainee trainee:trainees) {

String gen=trainee.getGender();

**if**(gen.equals(gender)) {

foundTrainees[i]=trainee;

i++;

}

}

**return** foundTrainees;

}

**public** Trainee[] getTrainees(**int** age) {

Trainee[] foundTrainees=**new** Trainee[0];

**int** i=0;

**for** (Trainee trainee:trainees) {

**int** ag=trainee.getAge();

**if**(ag==(age)) {

foundTrainees[i]=trainee;

i++;

}

}

**return** foundTrainees;

}

}

**package** com.css.corejava;

**public** **class** Trainee {

**private** **int** traineeID;

**private** String traineeName;

**private** **int** contactNUM;

**private** String emailID;

**private** String gender;

**private** **int** age;

@Override

**public** String toString() {

**return** "Trainee [traineeID=" + traineeID + ", traineeName=" + traineeName + ", contactNUM=" + contactNUM

+ ", emailID=" + emailID + ", gender=" + gender + ", age=" + age + "]";

}

**public** Trainee(**int** traineeID, String traineeName, **int** contactNUM, String emailID, String gender, **int** age) {

**super**();

**this**.traineeID = traineeID;

**this**.traineeName = traineeName;

**this**.contactNUM = contactNUM;

**this**.emailID = emailID;

**this**.gender = gender;

**this**.age = age;

}

**public** Trainee() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** **int** getTraineeID() {

**return** traineeID;

}

**public** **void** setTraineeID(**int** traineeID) {

**this**.traineeID = traineeID;

}

**public** String getTraineeName() {

**return** traineeName;

}

**public** **void** setTraineeName(String traineeName) {

**this**.traineeName = traineeName;

}

**public** **int** getContactNUM() {

**return** contactNUM;

}

**public** **void** setContactNUM(**int** contactNUM) {

**this**.contactNUM = contactNUM;

}

**public** String getEmailID() {

**return** emailID;

}

**public** **void** setEmailID(String emailID) {

**this**.emailID = emailID;

}

**public** String getGender() {

**return** gender;

}

**public** **void** setGender(String gender) {

**this**.gender = gender;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

}

**package** com.css.corejava;

**import** java.util.Arrays;

**public** **class** Main {

**public** **static** **void** main (String[] args) {

Trainee[] trainees=**new** Trainee[3];

trainees[0]=**new** Trainee(123, "raj",123 ,"gmail","Male",21);

trainees[1]=**new** Trainee(124, "priya", 124,"gmail","FeMale",20);

trainees[2]=**new** Trainee(125, "lokesh",125 ,"gmail","Male",21);

Batch batch=**new** Batch();

System.***out***.println(Arrays.*toString*(batch.getTrainees(21)));

System.***out***.println(Arrays.*toString*(batch.getTrainees("Male")));

}

}

1. Follow the given instructions and create an application using Java. (i) Create an entity class named Project with member variables as projectId, projectName, projectHead, noOfResources. (ii) Create an object for the Project class and through setters assign the values for all the member variables. (iii) Print the corresponding object.

**package** com.css.corejava.main;

**public** **class** Project {

**private** **int** projectId;

**private** String projectName;

**private** String projectHead;

**private** **int** noOfResources;

**public** Project(**int** projectId, String projectName, String projectHead, **int** noOfResources) {

**super**();

**this**.projectId = projectId;

**this**.projectName = projectName;

**this**.projectHead = projectHead;

**this**.noOfResources = noOfResources;

}

**public** **int** getProjectId() {

**return** projectId;

}

**public** **void** setProjectId(**int** projectId) {

**this**.projectId = projectId;

}

**public** String getProjectName() {

**return** projectName;

}

**public** **void** setProjectName(String projectName) {

**this**.projectName = projectName;

}

**public** String getProjectHead() {

**return** projectHead;

}

**public** **void** setProjectHead(String projectHead) {

**this**.projectHead = projectHead;

}

**public** **int** getNoOfResources() {

**return** noOfResources;

}

**public** **void** setNoOfResources(**int** noOfResources) {

**this**.noOfResources = noOfResources;

}

}

**package** com.css.corejava.main;

**public** **class** ProjectMain {

**public** **static** **void** main(String[] args) {

Project myProject=**new** Project(261, "Aruba", "Gopi", 15 );

System.***out***.println("Project Id is: "+myProject.getProjectId());

System.***out***.println("Name of the Project: "+myProject.getProjectName());

System.***out***.println("Name of the Project Head: "+myProject.getProjectHead());

System.***out***.println("Number of Resourses: "+myProject.getNoOfResources());

}

}

1. Prepare a StringServiceProvider class which has the following methods (a) To reverse a given string (b) To do linear search in a given string (c) To do search and replace operation in a given string Note: code the requirement with 2 possibilities (with and without static methods)

**package** com.css.corejava.strings;

**public** **class** StringServiceProvider {

**public** **static** **void** main(String[] args) {

String string="Chitrarasu";

**char** key='a';

**int** k;

**int** i=0;

**int** len=0;

**char** searchElement='a';

**char** replaceElement='A';

**try** {

**while**(**true**) {

**if**(string.charAt(i)!='\0')

len++;

i++;

}

}

**catch**(Exception e)

{

System.***out***.println("Length of the String:"+len);

}

System.***out***.println("Reversed String:");

**for**(**int** j=len-1;j>=0;j--) {

System.***out***.print(string.charAt(j));

}

System.***out***.println("");

**int** count=0;

**for**(k=0;k<len-1;k++) {

**if**(string.charAt(k)==key)

{

System.***out***.println("Searched Element:");

System.***out***.println("The searching word "+key+" found at the index of "+k);

count++;

}

}

**if**(count==0) {

System.***out***.println("The given word is not found");

}

StringServiceProvider s= **new** StringServiceProvider();

String replaced = s.replaced(string,searchElement,replaceElement);

System.***out***.println("After Replaced : "+replaced);

}

**public** String replaced(String name,**char** searchElement,**char** replaceElement) {

**int** len=6;

**char**[] arr=name.toCharArray();

**for**(**int** i=0;i<len;i++) {

**if**(arr[i]==searchElement)

{

arr[i]=replaceElement;

}

}

String replaced=**new** String(arr);

**return** replaced;

}

}

1. Follow the given instructions and create an application using Java. (i) Create a ‘BankAccount’ class with 3 data members, accountNo[use String], accountName and balance. (ii) Overload the BankAccount constructor to accept only accountNo and accountName variables. (iii) Initialize the balance with 1000.

**package** com.css.corejava.overload;

**import** java.util.Arrays;

**public** **class** BankAccount {

**private** String accountNo;

**private** String accountName;

**private** **double** balance=1000;

**private** BankAccount[] bankAccount;

**public** BankAccount[] getBankAccount() {

**return** bankAccount;

}

**public** **void** setBankAccount(BankAccount[] bankAccount) {

**this**.bankAccount = bankAccount;

}

**public** String getAccountNo() {

**return** accountNo;

}

**public** **void** setAccountNo(String accountNo) {

**this**.accountNo = accountNo;

}

**public** String getAccountName() {

**return** accountName;

}

**public** **void** setAccountName(String accountName) {

**this**.accountName = accountName;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** BankAccount(String accountNo, String accountName) {

**super**();

**this**.accountNo = accountNo;

**this**.accountName = accountName;

}

**public** BankAccount() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "BankAccount [accountNo=" + accountNo + ", accountName=" + accountName + ", balance=" + balance

+ ", Accounts=" + "]";

}

}

**package** com.css.corejava.overload;

**import** java.util.Arrays;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

BankAccount Acc= **new** BankAccount("123456789","chitrarasu");

System.***out***.println("Account Holder name: "+Acc.getAccountName());

System.***out***.println("Account Number: "+Acc.getAccountNo());

System.***out***.println("Account Balance: "+Acc.getBalance());

}

}

5. Follow the given instructions and create an application using Java.   
(i) Create a Bank Class having an array of BankAccount as a data member, populate the array through setters.  
 (ii) Introduce a static variable called lastAssignedNo(integer). This should be initialized to 0 in the beginning. While creating new bank accounts, the accountNo variable should not be supplied in the constructor parameter list. Instead it has to be computed as (lastAssignedNo + 1). Also modify the lastAssignedNo after creating a bank account.   
(iii) Create an IBankServiceProvider interface and declare the following methods:   
(a) BankAccount checkAccount(String accountNo) This checks whether the given account number is available in the array or not. If exists, it should return the object of BankAccount class, else return null. Reuse this method in all the other methods given below.   
(b) double getBalance( BankAccount account) This will return the balance in an account for the given account  
 (c) boolean depositMoney(BankAccount account, double amount) This deposits the given amount into the given account number after verifying whether the given account is present in the array or not.  
 (d) boolean withdrawMoney(BankAccount account, double amount) This will withdraw the given amount from the given account after verifying the existence of account as well as balance.   
(e) boolean transferMoney(BankAccount fromAccount, BankAccount toAccount amount) This transfers the money from one account to another account after verifying both the accounts are existing or not as well as balance of the ‘fromAccount’.   
(iv) Bank class should implement IBankServiceProvider interface and override the methods of the interfac

package bank;

import java.util.Arrays;

import bank.InvalidAccountNoException;

import bank.InvalidFundException;

import bank.IBankServiceProvider;

public class Bank implements IBankServiceProvider {

public BankAccount[] accounts;

private String IFSC;

private String bankName;

public Bank(BankAccount[] accounts,String ifsc,String bankName) {

super();

this.accounts=accounts;

IFSC=ifsc;

this.bankName=bankName;

}

public Bank() {

super();

}

public BankAccount CheckAccount(String accountNo)throws InvalidAccountNoException {

BankAccount foundAccount=null;

for(BankAccount account:accounts)

{

if(account.getAccountNo().equals(accountNo)) {

foundAccount=account;

break;

}

}

if(foundAccount==null) {

throw new InvalidAccountNoException();

}

return foundAccount;

}

public double getBalance(String accountNo)throws InvalidAccountNoException {

double balance=0.0;

BankAccount foundAccount=CheckAccount(accountNo);

balance=foundAccount.getBalance();

return balance;

}

public boolean depositMoney(String accountNo,double amount)throws InvalidAccountNoException{

boolean flagDeposit=false;

BankAccount foundAccount=CheckAccount(accountNo);

if(foundAccount!=null)

{

foundAccount.setBalance(foundAccount.getBalance()+amount);;

flagDeposit=true;

}

return true;

}

public boolean withdrawMoney(String accountNo,double amount)throws InvalidAccountNoException,InvalidFundException{

boolean flagWithdraw=false;

BankAccount foundAccount=CheckAccount(accountNo);

if(foundAccount!=null)

{

if(foundAccount.getBalance()>amount) {

foundAccount.setBalance(foundAccount.getBalance()-amount);;

flagWithdraw=true;

}

else {

throw new InvalidFundException();

}

}

return flagWithdraw;

}

public boolean transferMoney(String fromAccountNo,String toAccountNo,double amount)throws InvalidAccountNoException,InvalidFundException{

boolean transferFlag=false;

boolean flagWithdraw=withdrawMoney(fromAccountNo,amount);

boolean flagDeposit=depositMoney(toAccountNo,amount);

if(flagWithdraw) {

flagDeposit=depositMoney(toAccountNo,amount);

}

if(flagDeposit)

transferFlag=true;

else

depositMoney(fromAccountNo,amount);

return transferFlag;

}

public BankAccount[] getAccounts() {

return accounts;

}

public void setAccounts(BankAccount[] accounts) {

this.accounts = accounts;

}

public String getIFSC() {

return IFSC;

}

public void setIFSC(String iFSC) {

IFSC = iFSC;

}

public String getBankName() {

return bankName;

}

public void setBankName(String bankName) {

this.bankName = bankName;

}

@Override

public String toString() {

return "Bank [accounts=" + Arrays.toString(accounts) + ", IFSC=" + IFSC + ", bankName=" + bankName + "]";

}

}

**package** bank;

**public** **class** BankAccount {

**private** **static** **int** *lastAssignedNo*;

**static** {

*lastAssignedNo*=1000;

}

**private** String accountNo;

**public** String accountName;

**private** **double** balance=0;

**public** String getAccountNo() {

**return** accountNo;

}

**public** **void** setAccountNo(String accountNo) {

**this**.accountNo = accountNo;

}

**public** String getAccountName() {

**return** accountName;

}

**public** **void** setAccountName(String accountName) {

**this**.accountName = accountName;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

**public** BankAccount(String accountNo, String accountName, **double** balance) {

**super**();

**this**.accountNo = accountNo;

**this**.accountName = accountName;

**this**.balance = balance;

}

**public** BankAccount(String accountNo, String accountName) {

**super**();

**this**.accountNo = accountNo;

**this**.accountName = accountName;

}

**public** BankAccount() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "BankAccount [accountNo=" + accountNo + ", accountName=" + accountName + ", balance=" + balance + "]";

}

}

package bank;

import bank.Bank;

import bank.BankAccount;

import bank.InvalidAccountNoException;

import bank.InvalidFundException;

public class BankAccountMain {

public static void main(String[] args) throws InvalidAccountNoException {

// TODO Auto-generated method stub

BankAccount[] myobj =new BankAccount[3];

myobj[0]=new BankAccount("100","Hari");

myobj[1]=new BankAccount("101","Mani");

myobj[2]=new BankAccount("102","Harish");

Bank hdfcBank=new Bank(myobj,"AshokNagar","HDFC1234");

BankAccount foundAccount=null;

double balance=0.0;

try {

foundAccount=hdfcBank.CheckAccount("100");

balance=hdfcBank.getBalance("100");

hdfcBank.depositMoney("100",1500);

hdfcBank.withdrawMoney("100",1200);

hdfcBank.transferMoney("100","102",18);

}

catch(InvalidAccountNoException | InvalidFundException e) {

e.printStackTrace();

}

System.out.println(foundAccount.toString());

try {

// System.out.println("Output by Try and catch");

System.out.println(hdfcBank.getBalance("100"));

}

catch(InvalidAccountNoException e) {

e.printStackTrace();

}

System.out.println(balance);

System.out.println(hdfcBank.getBalance("102"));

}

}

package bank;

import bank.BankAccount;

import bank.Bank;

import bank.InvalidAccountNoException;

import bank.InvalidFundException;

public interface IBankServiceProvider {

BankAccount CheckAccount(String accountNo)throws InvalidAccountNoException;

double getBalance(String accountNo)throws InvalidAccountNoException;

boolean depositMoney(String accountNo,double amount)throws InvalidAccountNoException;

boolean withdrawMoney(String accountNo,double amount)throws InvalidAccountNoException,InvalidFundException;

boolean transferMoney(String fromAccount,String toAccountNo,double amount)throws InvalidAccountNoException,InvalidFundException;

}

package bank;

import bank.BankAccount;

import bank.Bank;

public class InvalidAccountNoException extends Exception {

private String errorMsg="Invalid Account Number";

public InvalidAccountNoException(String errorMsg) {

super();

this.errorMsg = errorMsg;

}

public InvalidAccountNoException() {

super();

// TODO Auto-generated constructor stub

}

public String getMessage() {

return this.errorMsg;

}

}

package bank;

import bank.BankAccount;

import bank.Bank;

public class InvalidFundException extends Exception {

private String errorMsg="Insufficient Balance";

public InvalidFundException(String errorMsg) {

super();

this.errorMsg = errorMsg;

}

public InvalidFundException() {

super();

// TODO Auto-generated constructor stub

}

public String getMessage() {

return this.errorMsg;

}

}

7. Design a class Company which has the following attributes: Company ID Address EmployeeMap < id, Employee> → Employee is an Entity class → Map is EmployeeMap → Company class should implement ICompanyserviceprovider to facilitate basic CRUD operations on Employee Class. ( Consider using a List over the EmployeeMap and update the necessary changes in the code )

**package** com.css.corejava.company;

**public** **class** Company {

**private** **int** companyID;

**private** Address address;

**private** EmployeeMap<Integer,String>employees =**new** Employee();

**public** **int** getCompanyID() {

**return** companyID;

}

**public** **void** setCompanyID(**int** companyID) {

**this**.companyID = companyID;

}

**public** Address getAddress() {

**return** address;

}

**public** **void** setAddress(Address address) {

**this**.address = address;

}

**public** EmployeeMap<Integer, String> getEmployees() {

**return** employees;

}

**public** **void** setEmployees(EmployeeMap<Integer, String> employees) {

**this**.employees = employees;

}

**public** Company(**int** companyID, Address address, EmployeeMap<Integer, String> employees) {

**super**();

**this**.companyID = companyID;

**this**.address = address;

**this**.employees = employees;

}

**public** Company() {

**super**();

// **TODO** Auto-generated constructor stub

}

@Override

**public** String toString() {

**return** "Company [companyID=" + companyID + ", address=" + address + ", employees=" + employees + "]";

}

}

**package** com.css.corejava.company;

**public** **class** Address {

**private** **int** houseNo;

**private** String streetName;

**private** String city;

**private** String state;

**private** **int** pinCode;

**public** Address(**int** houseNo, String streetName, String city, String state, **int** pinCode) {

**super**();

**this**.houseNo = houseNo;

**this**.streetName = streetName;

**this**.city = city;

**this**.state = state;

**this**.pinCode = pinCode;

}

**public** Address() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** **int** getHouseNo() {

**return** houseNo;

}

**public** **void** setHouseNo(**int** houseNo) {

**this**.houseNo = houseNo;

}

**public** String getStreetName() {

**return** streetName;

}

**public** **void** setStreetName(String streetName) {

**this**.streetName = streetName;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** String getState() {

**return** state;

}

**public** **void** setState(String state) {

**this**.state = state;

}

**public** **int** getPinCode() {

**return** pinCode;

}

**public** **void** setPinCode(**int** pinCode) {

**this**.pinCode = pinCode;

}

@Override

**public** String toString() {

**return** "Address [houseNo=" + houseNo + ", streetName=" + streetName + ", city=" + city + ", state=" + state

+ ", pinCode=" + pinCode + "]";

}

}