**DAY-4 ASSESMENT**

**BASICS OF JAVA**

**Problem-1**

BankAccount Class

**package** org.software.com;

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

**int** accountNo;

String custName;

String accType;

**float** balance;

**private** **static** **final** **float** ***minSaving*** = 1000.0f;

**private** **static** **final** **float** ***minCurrent*** = 5000.0f;

**public** BankAccount(**int** accountNo, String custName, String accType, **float** balance) {

**super**();

**this**.accountNo = accountNo;

**this**.custName = custName;

**this**.accType = accType;

**this**.balance = balance;

isValidBalance();

}

**void** isValidBalance() {

**if** (**this**.balance < 0) {

**throw** **new** NegativeAmountException("Negative Amount");

} **else** **if** (accType.equalsIgnoreCase("Current") && balance < ***minCurrent***) {

**throw** **new** LowBalanceException("Low Balance");

} **else** **if** (accType.equalsIgnoreCase("Savings") && balance < ***minSaving***) {

**throw** **new** LowBalanceException("Low Balance");

}

}

**public** **void** deposit(**float** amount) {

**if** (amount < 0)

**throw** **new** NegativeAmountException("Negative Amount");

**this**.balance += amount;

}

**public** **float** getBalance() {

isValidBalance();

**return** **this**.balance;

}

}

NegativeAmountException Class

**class** NegativeAmountException **extends** RuntimeException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** NegativeAmountException(String message) {

**super**("NegativeAmountException : " + message);

}

}

LowBalanceException Class

**class** LowBalanceException **extends** RuntimeException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** LowBalanceException(String message) {

**super**("LowBalanceException : " + message);

}

}

BankApplication Class

**package** org.software.com;

**import** java.util.Scanner;

**public** **class** BankApplication {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter Account Number");

**int** accountNo = sc.nextInt();

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

String name = sc.next();

System.***out***.println("Enter Account Type");

String accountType = sc.next();

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

**float** amount = sc.nextFloat();

**try** {

BankAccount account = **new** BankAccount(accountNo, name, accountType, amount);

account.deposit(100);

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

} **catch** (NegativeAmountException | LowBalanceException e) {

System.***out***.println(e.getMessage());

}

}

}

**Problem-2**

CricketRating Class

**package** org.software.com;

**public** **class** CricketRating {

String playerName;

**float** critic1;

**float** critic2;

**float** critic3;

**float** avgRating;

String coins;

**public** CricketRating(String playerName, **float** critic1) {

**super**();

**this**.playerName = playerName;

**this**.critic1 = critic1;

**this**.avgRating = critic1;

calculateCoin();

}

**public** CricketRating(String playerName, **float** critic1, **float** critic2) {

**super**();

**this**.playerName = playerName;

**this**.critic1 = critic1;

**this**.critic2 = critic2;

**this**.calculateAvarageRating(critic1, critic2);

}

**public** CricketRating(String playerName, **float** critic1, **float** critic2, **float** critic3) {

**super**();

**this**.playerName = playerName;

**this**.critic1 = critic1;

**this**.critic2 = critic2;

**this**.critic3 = critic3;

**this**.calculateAvarageRating(critic1, critic2, critic3);

}

**void** calculateAvarageRating(**float** critic1, **float** critic2) {

**this**.avgRating = (critic1 + critic2) / 2;

calculateCoin();

}

**void** calculateAvarageRating(**float** critic1, **float** critic2, **float** critic3) {

**this**.avgRating = (critic1 + critic2 + critic3) / 3;

calculateCoin();

}

**private** String calculateCoin() {

**if** (avgRating >= 7) {

coins = "Gold";

**return** coins;

} **else** **if** (avgRating < 7 && avgRating >= 5) {

coins = "Silver";

**return** coins;

} **else** **if** (avgRating < 5 && avgRating >= 2) {

coins = "Cooper";

**return** coins;

} **else** {

**throw** **new** NotElegibleException("Not Eligible");

}

}

**void** display() {

System.***out***.println(**this**.playerName + " " + **this**.avgRating + " " + **this**.coins);

}

}

NotElegibleException Class

**class** NotElegibleException **extends** RuntimeException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** NotElegibleException(String message) {

**super**("NotElegibleException : " + message);

}

}

TestRating Class

**package** org.software.com;

**import** java.util.Scanner;

**public** **class** TestRating {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter name of The Player");

String playerName = sc.nextLine();

System.***out***.println("How many critics do you have: ");

**int** critics = sc.nextInt();

CricketRating cr;

**float** critic1;

**float** critic2;

**float** critic3;

**try** {

**switch** (critics) {

**case** 1:

System.***out***.println("Enter your critic's rating:");

critic1 = sc.nextFloat();

cr = **new** CricketRating(playerName, critic1);

cr.display();

**break**;

**case** 2:

System.***out***.println("Enter your first critic's rating:");

critic1 = sc.nextFloat();

System.***out***.println("Enter your second critic's rating:");

critic2 = sc.nextFloat();

cr = **new** CricketRating(playerName, critic1, critic2);

cr.display();

**break**;

**case** 3:

System.***out***.println("Enter your first critic's rating:");

critic1 = sc.nextFloat();

System.***out***.println("Enter your second critic's rating:");

critic2 = sc.nextFloat();

System.***out***.println("Enter your third critic's rating:");

critic3 = sc.nextFloat();

cr = **new** CricketRating(playerName, critic1, critic2, critic3);

cr.display();

**break**;

**default**:

**throw** **new** NotElegibleException("Not Eligible");

}

} **catch** (NotElegibleException e) {

System.***out***.println(e.getMessage());

} **finally** {

sc.close();

}

}

}

**Problem-3**

Applicant Class

**package** org.software.com;

**public** **class** Applicant {

String applicantName;

String postApplied;

**int** applicantAge;

**public** Applicant(String applicantName, String postApplied, **int** applicantAge) {

**super**();

**this**.applicantName = applicantName;

**this**.postApplied = postApplied;

**this**.applicantAge = applicantAge;

}

}

CatheyBankException Class

**class** CatheyBankException **extends** RuntimeException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** CatheyBankException(String message) {

**super**(message);

}

}

Validator Class

**package** org.software.com;

**public** **class** Validator {

**public** **void** validApplicant(Applicant applicant) {

**if** (isValidApplicantName(applicant.applicantName)) {

**if** (isValidPost(applicant.postApplied)) {

**if** (isValidAge(applicant.applicantAge)) {

System.***out***.println("All Values Are Valid");

} **else** {

**throw** **new** InvalidAgeException("Invalid Age");

}

} **else** {

**throw** **new** InvalidPostException("invalid Post");

}

} **else** {

**throw** **new** InvalidNameException("Invalid Applicant Name");

}

}

**private** **boolean** isValidAge(**int** applicantAge) {

**if** (applicantAge >= 18 && applicantAge <= 30)

**return** **true**;

**return** **false**;

}

**private** **boolean** isValidPost(String postApplied) {

**if** (postApplied.equalsIgnoreCase("Probationary Officers") || postApplied.equalsIgnoreCase("Assistants")

|| postApplied.equalsIgnoreCase("Special Cadre Officer"))

**return** **true**;

**return** **false**;

}

**public** **boolean** isValidApplicantName(String name) {

**return** name != **null** && !name.isEmpty();

}

}

InvalidNameException Class

**class** InvalidNameException **extends** CatheyBankException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** InvalidNameException(String message) {

**super**("InvalidNameException : " +message);

}

}

InvalidPostException Class

**class** InvalidPostException **extends** CatheyBankException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** InvalidPostException(String message) {

**super**("InvalidPostException : " +message);

}

}

InvalidAgeException Class

**class** InvalidAgeException **extends** CatheyBankException {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** InvalidAgeException(String message) {

**super**("InvalidAgeException : " +message);

}

}

ApplicationPage Class

**package** org.software.com;

**import** java.util.Scanner;

**public** **class** ApplicationPage {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter Name :");

String name = sc.nextLine();

//Assistants or Probationary Officers or Special Cadre Officer

System.***out***.println("Enter Post Name :");

String Post = sc.nextLine();

System.***out***.println("Enter age : ");

**int** age = sc.nextInt();

Applicant applicant=**new** Applicant(name, Post, age);

//Applicant applicant = new Applicant("", "Assistants", 30);

Validator v=**new** Validator();

**try** {

v.validApplicant(applicant);

} **catch** (CatheyBankException e) {

System.***out***.println(e.getMessage());

}

}

}