10-04-2023 HomeWork

1.

```
import java.util.*;
public class Main
{
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		try{
		    int a = sc.nextInt();
		    int b = sc.nextInt();
		    System.out.println(a/b);
		}
		catch(ArithmeticException e){
		    System.out.println(e);
		}
	}
}
```

2.

```
public class Main {
   public static void main(String[] args) {

      String str = null;

      try {

         char c = str.charAt(0);
         System.out.println(c);
      } catch (NullPointerException e) {

         System.out.println(e.getClass().getName());
      }
   }
}
```

3.

```
import java.util.*;

public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      try {
```

```
            int input = scanner.nextInt();

            System.out.println(input);
        } catch (InputMismatchException e) {
            System.out.println(e);
        }
    }
}
```

4.

```
import java.util.*;

class DotException extends Exception {
    public DotException(String message) {
        super(message);
    }
}

class AtTheRateException extends Exception {
    public AtTheRateException(String message) {
        super(message);
    }
}

class DomainException extends Exception {
    public DomainException(String message) {
        super(message);
    }
}

class EmailValidation {
    private String email;

    public EmailValidation(String email) {
        this.email = email;
    }

    public void checkEmail() throws DotException, AtTheRateException, DomainException {
        if (!email.contains(".")) {
            throw new DotException("DotException: Invalid Dot usage");
        }
        if (!email.contains("@")) {
```

```
            throw new AtTheRateException("AtTheRateException: Invalid @ usage");
        }

        String[] parts = email.split("@");
        if (parts.length != 2) {
            throw new AtTheRateException("AtTheRateException: Invalid @ usage");
        }

        String local = parts[0];
        String domain = parts[1];

        if (local.isEmpty() || domain.isEmpty()) {
            throw new AtTheRateException("AtTheRateException: Invalid @ usage");
        }

        String[] validDomains = {"in", "com", "net", "biz"};
        boolean isValidDomain = false;
        for (String valid : validDomains) {
            if (domain.equals(valid)) {
                isValidDomain = true;
                break;
            }
        }

        if (!isValidDomain) {
            throw new DomainException("DomainException: Invalid Domain");
        }

        System.out.println("Valid email address: " + email);
    }
}

public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String email = sc.nextLine();

        EmailValidation em = new EmailValidation(email);
        try {
            em.checkEmail();
        } catch (DotException | AtTheRateException | DomainException e) {
            System.out.println("Invalid email address");
            System.out.println(e.getMessage());
        }
```

```
    }
}

5.
import java.util.Scanner;

class InvalidRegisterNumberException extends Exception {
    public InvalidRegisterNumberException(String message) {
        super(message);
    }
}

class InvalidMobileNumberException extends Exception {
    public InvalidMobileNumberException(String message) {
        super(message);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        try {
            String rno = sc.nextLine();
            String phno = sc.nextLine();

            validateRegisterNumber(rno);
            validateMobileNumber(phno);

            System.out.println("Valid");
        } catch (InvalidRegisterNumberException | InvalidMobileNumberException e) {
            System.out.println("Invalid");
            System.out.println(e.getMessage());
        }
    }

    private static void validateRegisterNumber(String rno) throws
InvalidRegisterNumberException {
        if (rno.length() != 9) {
            throw new InvalidRegisterNumberException("IllegalArgumentException - Register
Number does not contain exactly 9 characters");
        }

        String firstTwo = rno.substring(0, 2);
```

```
        String middleThree = rno.substring(2, 5);
        String lastFour = rno.substring(5);

        for (int i = 0; i < firstTwo.length(); i++) {
            if (!Character.isDigit(firstTwo.charAt(i))) {
                throw new InvalidRegisterNumberException("IllegalArgumentException - First two
characters of Register Number should be digits");
            }
        }
        for (int i = 0; i < middleThree.length(); i++) {
            if (!Character.isLetter(middleThree.charAt(i))) {
                throw new InvalidRegisterNumberException("IllegalArgumentException - Middle three
characters of Register Number should be alphabets");
            }
        }
        for (int i = 0; i < lastFour.length(); i++) {
            if (!Character.isDigit(lastFour.charAt(i))) {
                throw new InvalidRegisterNumberException("IllegalArgumentException - Last four
characters of Register Number should be digits");
            }
        }
    }

    private static void validateMobileNumber(String phno) throws InvalidMobileNumberException
{
        if (phno.length() != 10) {
            throw new InvalidMobileNumberException("IllegalArgumentException - Mobile Number
does not contain exactly 10 characters");
        }

        char[] phnoChars = phno.toCharArray();
        for (char c : phnoChars) {
            if (!Character.isDigit(c)) {
                throw new InvalidMobileNumberException("NumberFormatException - Mobile Number
cannot contain any character other than a digit");
            }
        }
    }
}
```