CENTRO UNIVERSITARIO UAEM ATLACOMULCO

CARRERA

LICENCIATURA EN INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA

PARADIGMAS DE LA PROGRAMACIÓN

DOCENTE

JULIO ALBERTO DE LA TEJA LÓPEZ

ALUMNO

ALEXIS VALENCIA MARTÍNEZ

FECHA DE ENTREGA

4-SEPTIEMBRE-2023

➢ FIGURE.JAVA

```java
public abstract class Figure{
    //Name of figure
    protected String Name;
    public Figure(String Name){
        this.Name= Name;
    }
    //methods
    public abstract double calculateArea();
    public abstract double calculatePerim();
}
```

➢ RECTANGULE.JAVA

```java
public class Rectangle extends Figure{
    private double base;
    private double length;

    public Rectangle(double base, double length){
        super("Rectangle");
        this.length= length;
        this.base = base;
    }
    @Override
    //Area calculate
    public double calculateArea() {
        return base * length;
    }
    //Perimeter calculate
    public double calculatePerim(){
        return(base+length);
    }

}
```

➢ TRIANGLE.JAVA

```java
public class Triangle extends Figure{
    //Add base, height and sides
    private double base;
    private double height;
    private double sideA;
    private double sideB;
    private double hypotnuse;
    public Triangle(double base, double height,double sideA,double sideB,double
hypotnuse)throws Exception{
        super("Triangle");
        //Failure to comply will result in an error
        if (sideA + sideB <= hypotnuse || sideA + sideB <= hypotnuse || sideB + hypotnuse <=
sideA) {
            throw new Exception("Error: re-enter the sides ");
```

```java
        }
        this.base = base;
        this.height= height;
        this.sideA= sideA;
        this.sideB= sideB;
        this.hypotnuse= hypotnuse;
    }
    @Override
    //Area calculate
    public double calculateArea() {
        double area = 0;
        try {
            area = (base * height) / 2;
        } catch (ArithmeticException e) {
            System.out.println("Error: División entre cero");
        }
        return area;
    }
    //perimeter calculate
    public double calculatePerim(){
        return(sideA+sideB+hypotnuse);
    }

}
```

➢ **CIRCLE.JAVA**

```java
public  class Circle extends Figure{
    private double radius;
    public Circle(double radius){
        super("Circle");
        this.radius=radius;
    }
    @Override
    //Area calculate
    public double calculateArea() {
        return Math.PI * Math.pow(radius, 2);
    }
    //perimeter calculate
    public double calculatePerim(){
        return Math.PI*(radius+radius);
    }

}
```
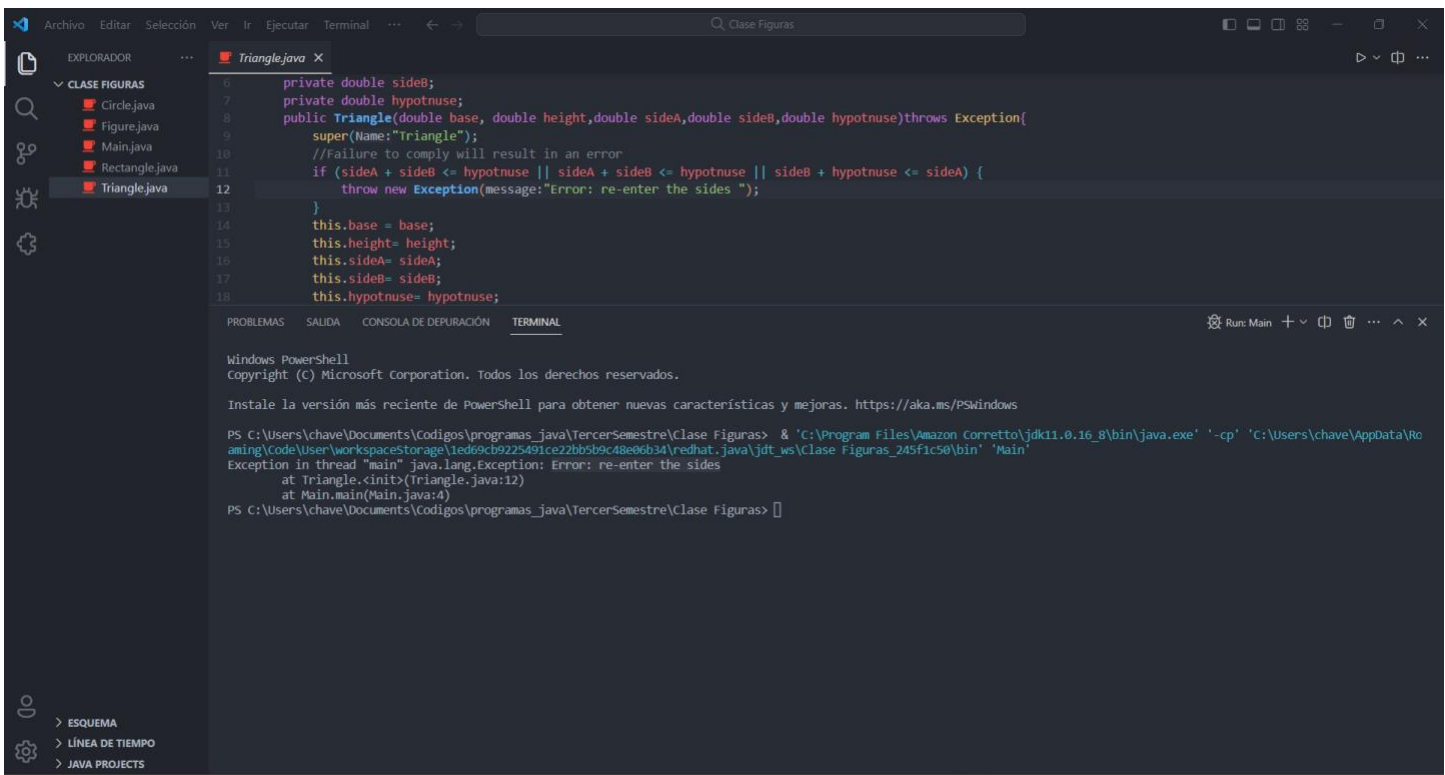
```java
public class Main {
    public static void main(String[] args) throws Exception {
        Rectangle rectangle = new Rectangle(4.0, 5.2);
        Triangle triangle = new Triangle(10, 10,0,0,10);
        Circle circle = new Circle(10);

        System.out.println("Area of "+rectangle.Name + ": "+
rectangle.calculateArea()+"  Perimeter: "+ rectangle.calculatePerim());
        System.out.println("Area of "+triangle.Name + ": "+
triangle.calculateArea()+"  Perimeter: "+ triangle.calculatePerim());
        System.out.println("Area of "+circle.Name + ": "+
circle.calculateArea()+"  Perimeter: "+ circle.calculatePerim());
    }
}
```

- **In which figure can this error occur?**
  triangle would be the closest
- **How did you solve it?**
  Adding an exception that finds out when it divides by 0

➢ **SALIDA**

➢ **CHARACTER.JAVA**

```java
public  class  Character {
    protected String Name;
    protected int Level;
    public Character(String Name, int level){
        if (level < 1) {
            // Error in case of level is minor than 1
            throw new IllegalArgumentException("Level is minor than 1");
        }
        this.Name = Name;
        this.Level = level;
    }
    public void  Attack(int level){
        if (level < 5) {
            // Error in case of level is minor than 1
            throw new RuntimeException("Level is minor than 5, you cannot attack");
        }else{
            System.out.println("You are attacking!");
        }
    }
    public void displayAttributes() {
        System.out.println("Name: " + Name);
        System.out.println("Level: " + Level);
    }
}
```

➢ **ENEMY.JAVA**

```java
public class Enemy extends Character{
    protected String type;

    public Enemy(String Name, int Level,String Enemy){
        super("Boss Final", 99);
        this.type = Enemy;
    }
    public void Scream(){
        System.out.println(Name + "Scream!!!");
    }
    @Override
    public void displayAttributes(){
        super.displayAttributes();
        System.out.println("Type: "+ type);
    }
}
```

```java
public   class Player extends Character{
    protected String Class;
    public Player(String Name,int level,String Class){
        super("Jose", 12);
        this.Class= Class;
    }
   public void useAbility(Enemy enemy){
    System.out.println("Use special ability vs "+ enemy.Name);
   }

   @Override
   public void displayAttributes(){
    super.displayAttributes();
    System.out.println("Class: "+ Class);
   }
}
```

```java
import java.util.Random;

public class Main {
    public static void main(String[] args) {
        Player player = new Player("Jorge", 50, "Mago");
        Enemy Zombie = new Enemy("Zombie", 23, "Undead");
        Enemy Boss = new Enemy("Medusa", 90, "Boss");
        Enemy Boss_Final = new Enemy("Dragon", 90, "Boss Final");
        player.displayAttributes();
        player.Attack(50);

        Random random = new Random();
        int RandomX = random.nextInt(3);
        Enemy randomEnemy;

        if(RandomX==0){
            randomEnemy = Boss_Final;
        }else if(RandomX == 1){
            randomEnemy = Boss;
        }else{
            randomEnemy = Zombie;
        }
        player.useAbility(randomEnemy);

        System.out.println();
        Zombie.displayAttributes();
        Zombie.Attack(23);
        Zombie.Scream();

        System.out.println();
```

```
        Boss.displayAttributes();
        Boss.Attack(90);
        Boss.Scream();

        System.out.println();
        Boss_Final.displayAttributes();
        Boss_Final.Attack(2);
        Boss_Final.Scream();
    }
}
```

➢ **SALIDA**

```java
public  class palette{
    protected String Flavor;
    protected double price;

    public palette(String Flavor, double price){
        //In case it is marked that it is one of the two types, it will give an error
        if (!Flavor.equals("Water") && !Flavor.equals("Cream")) {
            throw new IllegalArgumentException("The flavor is Water or Cream");
        }
        this.Flavor= Flavor;
        this.price= price;
    }


    public void showInfo(){
        System.out.println("Flavor is: "+ Flavor);
        System.out.println("Price is: "+ price);
    }
    public void discount(double percentage){
        double discount= price* (percentage/100);
        price = price - discount;
        System.out.println("Discount Applied is: " + discount);
    }

}
```

➤ PCREAM.JAVA

```java
public class PCream extends palette{
    protected Boolean cream;
    public PCream(String Flavor,Double price,boolean cream){
        super("Crem", 30);
        this.cream= cream;
    }
    public void showCream(){
        System.out.println("Cream texture: "+(cream ? "Yes": "No"));
    }
    public void ChangePrice(){
        price= price +6;
    }
}
```

➤ PWATER.JAVA

```java
public class PWater extends palette{
    protected boolean BaseWater;
    public PWater(String flavor, double Price,boolean Base){
        super("Agua", 20.0);
        this.BaseWater = Base;
    }
```

```java
    public void showBaseW(){
        System.out.println("Water Base: "+(BaseWater ? "Yes": "No"));
    }
    public void ChangePrice(){
        price = price+2;
    }

}
```

➤ **MAIN.JAVA**

```java
import java.util.Random;

public class Main{
    public static void main(String[] args) {
        PWater waterP = new PWater("Lemon", 19.5, true);
        PCream creamP = new PCream("Milk", 35.5, true);

        boolean Applied;
        Random random = new Random();
        int randomX = random.nextInt(1);
        if(randomX==0){
            Applied= true;
        }else{
            Applied=false;
        }

        waterP.showInfo();
        waterP.showBaseW();
        waterP.ChangePrice();


        creamP.showInfo();
        creamP.showCream();
        creamP.ChangePrice();
        if(Applied == true){
            System.out.println("Congratulations!!!!!\n"+"You win discount in your purchase");
            waterP.discount(20);
            creamP.discount(10);
        }


    }
}
```
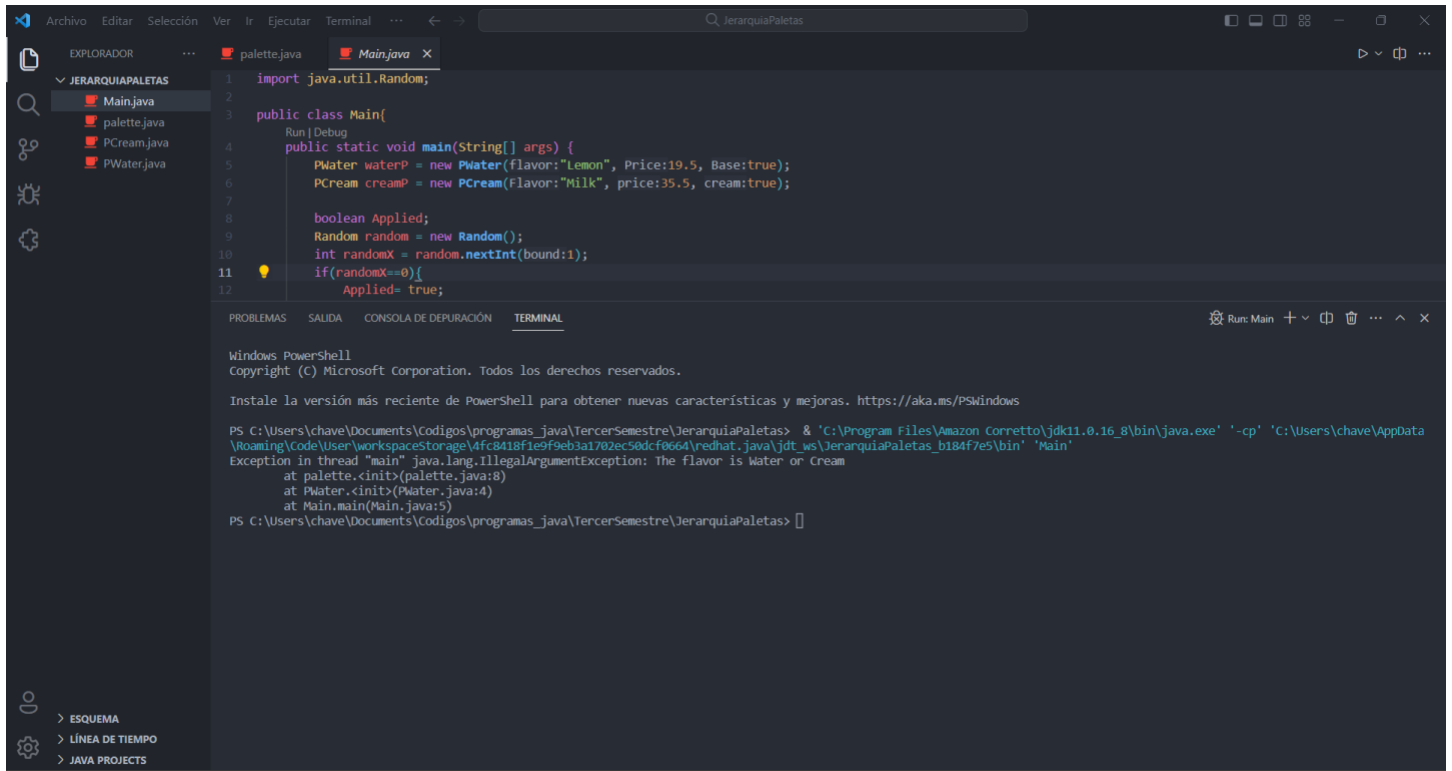
```java
import java.util.Random;

public class Main{
    Run | Debug
    public static void main(String[] args) {
        PWater waterP = new PWater(flavor:"Lemon", Price:19.5, Base:true);
        PCream creamP = new PCream(Flavor:"Milk", price:35.5, cream:true);

        boolean Applied;
        Random random = new Random();
        int randomX = random.nextInt(bound:1);
        if(randomX==0){
            Applied= true;
```

PROBLEMAS   SALIDA   CONSOLA DE DEPURACIÓN   TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\chave\Documents\Codigos\programas_java\TercerSemestre\JerarquiaPaletas>  & 'C:\Program Files\Amazon Corretto\jdk11.0.16_8\bin\java.exe' '-cp' 'C:\Users\chave\AppData
\Roaming\Code\User\workspaceStorage\4fc8418f1e9f9eb3a1702ec50dcf0664\redhat.java\jdt_ws\JerarquiaPaletas_b184f7e5\bin' 'Main'
Exception in thread "main" java.lang.IllegalArgumentException: The flavor is Water or Cream
        at palette.<init>(palette.java:8)
        at PWater.<init>(PWater.java:4)
        at Main.main(Main.java:5)
PS C:\Users\chave\Documents\Codigos\programas_java\TercerSemestre\JerarquiaPaletas> []
```

```
    +calculateArea()
    +calculatePerim()
}

class Circle {
    -radius: double
    +Circle(radius)
    +calculateArea()
    +calculatePerim()
}

class Main {
    +main(args)
}

Rectangle --|> Figure
Triangle --|> Figure
Circle --|> Figure
@enduml
```

Pure Javascript [NEW!] PNG SVG ASCII Art

```
class Enemy {
    -type: String
    +Enemy(Name, level, Enemy)
    +Scream()
    +displayAttributes()
}

class Player {
    -Class: String
    +Player(Name, level, Class)
    +useAbility(enemy)
    +displayAttributes()
}

class Main {
    +main(args)
}

Enemy --|> Character
Player --|> Character
@enduml
```

Pure Javascript [NEW!] PNG SVG ASCII Art

https://www.plantuml.com/plantuml/uml/SoWkllmgAStDuN8EllmkDZ1KiAdHrLM0S8oWWiOAMd0n4wYOgK8-NCmCAcQkeAS75RA02ba...

```
class PCream {
  -cream: Boolean
  +PCream(Flavor, price, cream)
  +showCream()
  +ChangePrice()
}

class PWater {
  -BaseWater: Boolean
  +PWater(flavor, Price, Base)
  +showBaseW()
  +ChangePrice()
}

class Main {
  +main(args)
}

PCream --|> palette
PWater --|> palette
@enduml
```

//www.plantuml.com/plantuml/png/VP6nQiD038PtFuN6AubVm4C92HGw50mTEgjdnJ6STKOdjqFRTwzFvvfwoIRz_oS-x3ibHT5v39KFc19C64YLuAi2g9y2VaHfu5 Decode URL

Submit   Pure Javascript NEW! PNG SVG ASCII Art

chat 47 online | online diagrams 274,299,066 | current rate 261 diag. per minute | peak rate 1004 diag. per minute

C PCream
- cream: Boolean
- PCream(Flavor, price, cream)
- showCream()
- ChangePrice()

C PWater
- BaseWater: Boolean
- PWater(flavor, Price, Base)
- showBaseW()
- ChangePrice()

C Main
- main(args)

C palette
- Flavor: String
- price: double
- palette(Flavor, price)
- showInfo()
- discount(percentage)