

```
1 Programming 'Dart:io' {
```

```
2  
3 [abstract Class  
4 clasesAbstractas]
```

```
5  
6  
7  
8 < Ximena Jaramillo Villada>  
9 < Juan Esteban Ramirez Jimenez>  
10 < Juan Sebastian Aristizabal>  
11 < Sebastian Ramirez Ramirez>  
12 < Yefry Alejandro Marin Serrano>
```

```
13 }  
14
```

Contents Of

'Definición Clases Abstractas ';

En Dart, las clases abstractas proporcionan un mecanismo para definir interfaces y comportamientos comunes sin implementarlos directamente. Estas clases sirven como modelos para otras clases que heredan de ellas.

Contents Of

'Definición Metodos Abstractos ';

Las clases abstractas pueden contener métodos abstractos, es decir, métodos que no tienen una implementación definida en la clase abstracta, sino que deben ser implementados por las clases que heredan de ella.

Ejemplo Of 'Clases abstractas'

```
1  abstract class Persona {
2      String nombre;
3      int edad;
4
5      Persona(this.nombre, this.edad);
6
7      void saludar() {
8          print('Hola, mi nombre es $nombre y tengo $edad años.');
```

```
9      }
10     void despedir();
11 }
12
13 class Hombre extends Persona {
14     Hombre(String nombre, int edad) : super(nombre, edad);
15
16     void despedir(){
17         print("Me voy porque tengo sueño");
18     }
19 }
20
21 void main() {
22     Hombre holaa = Hombre("xime", 25);
23     holaa.saludar();
24     holaa.despedir();
25 }
```

Contents Of 'Definición Mixins';

En Dart, los mixins permiten la reutilización de código de manera flexible mediante la composición en lugar de la herencia. Se definen con la palabra clave **mixin** y se pueden usar en otras clases con la palabra clave **with**, lo que promueve un código más limpio y modular

Ejemplo Of 'mixin' {

```
//Definición de un mixin
mixin MiMixin {
  // Definición de métodos, campos y getters/setters
  void metodoMixin() {
    print('Método del mixin');
  }

  int valorMixin = 42;
}

// Uso del mixin en una clase
class MiClase with MiMixin {
  // Cualquier otra definición de la clase
}

Run | Debug
void main() {
  // Creación de una instancia de la clase que utiliza el mixin
  var miObjeto = MiClase();

  // Uso de los métodos y campos del mixin
  miObjeto.metodoMixin(); // Imprime: Método del mixin
  print(miObjeto.valorMixin); // Imprime: 42
}
```

Contents Of 'abstract Class // Interfaces';

En Dart, las clases abstractas pueden contener tanto métodos abstractos como concretos y permiten la herencia de una sola clase, mientras que las interfaces solo definen métodos abstractos y permiten que las clases implementen múltiples interfaces. Ambos son útiles para establecer contratos y estructuras en el diseño de programas, pero se utilizan en contextos ligeramente diferentes.

Ejemplo Of 'interfaces'

```
// Definición de una interfaz
abstract class Volador {
  void volar();
}

// Clase que implementa la interfaz
class Pajaro implements Volador {
  @override
  void volar() {
    print('El pájaro está volando.');
```

Run | Debug

```
void main() {
  // Creación de una instancia de la clase que implementa la interfaz
  var pajaro = Pajaro();

  // Llamada al método de la interfaz
  pajaro.volar(); // Salida: El pájaro está volando.
}
```