

```

type Pc struct {
    nombre string
    nivel  int
}

```

```

func Pc1 () Pc {
    return Pc { "", 0 }
}

```

```

func Pc2 ( nombre string, nivel int) Pc {
    return Pc { nombre, nivel }
}

```

```

func Pc3 (nivel int) Pc {
    return Pc { "Steven", nivel }
}

```

```

func Pc4 (P.Pc) Pc {
    return Pc { P.nombre, P.nivel }
}

```

```

type Abstractola interface {
    Saludo ()
}

```

```

type persona struct {
    nombre string
    nivel  int
}

```

```

func persona1 () persona {
    return persona { "Steven", 10 }
}

```

```

func persona2 (nombre string, nivel int) persona {
    return persona { nombre, nivel }
}

```

```

func persona3 (nombre string) persona {
    return persona { nombre, 10 }
}

```

```

func persona4 (P persona) persona {
    return persona { P.nombre, P.nivel }
}

```


Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

Package main

```
type Persona interface {  
    jugar()  
}
```

Package main

```
type mesa interface {  
    peso()  
}
```

Package main

```
type silla interface {  
    peso()  
}
```

Package main

```
type pc interface {  
    peso()  
}
```

Package main

```
type Hoja interface {  
    peso()  
}
```



```

type Pc struct {
    nombre string
    nivel  int
}

```

```

func Pc1() Pc {

```

```

    return Pc{"Steven", 18}
}

```

```

func Pc2(nombre string, nivel int) Pc {
    return {nombre, nivel}
}

```

```

func Pc3(nombre string) Pc {
    return Pc{nombre, 18}
}

```

```

func Pc4(p Pc) Pc {

```

```

    return Pc{p.nombre, p.nivel}
}

```

```

type mesa struct {
    nombre string
    nivel  int
}

```

```

func mesa1() mesa {

```

```

    return Pc{"Steven", 18}
}

```

```

func mesa2(nombre string, nivel int) mesa {
    return {nombre, nivel}
}

```

```

func mesa3(nombre string) mesa {
    return mesa{nombre, 18}
}

```

```

func mesa4(m mesa) mesa {
    return mesa{m.nombre, m.nivel}
}

```



```

type Persona struct {
    nombre string
    nivel  int
}

```

```

func newPersona() Salud {
    return Salud {"", 0}
}

```

```

func newPersonaa (nombre string, nivel int) persona {
    return persona { nombre, nivel }
}

```

```

func newPersonaaa (int nivel) persona {
    return persona {"Steven", nivel}
}

```

```

func newPersonaaaa (p persona) persona {
    return persona {p.nombre, p.nivel}
}

```

```

type AbstractHola interface {
    Salud()
}

```

```

type mesa struct {
    nombre string
    nivel  int
}

```

```

func newMesa1() mesa {
    return mesa {"", 0}
}

```

```

func newMesa2 (nombre string, nivel int) mesa {
    return mesa {nombre, nivel}
}

```

```

func newMesa3 (int nivel) mesa {
    return mesa {"Steven", nivel}
}

```

```

func newMesa4 (m mesa) mesa {
    return mesa {m.nombre, m.nivel}
}

```

```

type abstractHola interface {
    Salud()
}

```


Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
public class mesa {
    String nombre;
    int nivel;

    public mesa () {
        nombre = "Stiven";
        nivel = 50;
    }

    public persona (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public persona (String nombre) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public persona (persona p) {
        this.nombre = p.nombre;
        this.nivel = p.nivel;
    }
}

public class silla {
    String nombre;
    int nivel;

    public silla () {
        nombre = "Stiven";
        nivel = 500;
    }

    public silla (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public silla (String nombre) {
        this.nombre = nombre;
        this.nivel = 10;
    }

    public silla (silla s) {
        this.nombre = s.nombre;
        this.nivel = s.nivel;
    }
}
```



```

public class PC {
    String nombre;
    int nivel;

    public PC () {
        nombre = "Stiven";
        int = 50;
    }

    public PC (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public PC (String nombre) {
        this.nombre = nombre;
        this.nivel = 99;
    }

    public PC (PC p) {
        this.nombre = p.nombre;
        this.nivel = p.nivel;
    }
}

```

```

public class Pcel {
    String nombre;
    int nivel;

    public Pcel () {
        nombre = "Lenovo";
        nivel = 2;
    }

    public Pcel (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public Pcel (String nombre) {
        this.nombre = nombre;
        this.nivel = 10;
    }

    public Pcel (Pcel p) {
        this.nombre = p.nombre;
        this.nivel = p.nivel;
    }
}

```


Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

Package Java;

interface Jogar {

void jogar();

}

Package Java;

interface Jogar1

void jogar1();

}

Package Java;

interface regla {

void mude();

}

Package Java;

interface silla {

void sopartar();

}

Package Java;

interface mesa {

void peso();

}


```
abstract class Persona {
```

```
    String nombre;  
    int nivel;
```

```
    public Hola () {}
```

```
    public Hola (String nombre, int nivel) {
```

```
        this.nombre = nombre;  
        this.nivel = nivel;
```

```
    }
```

```
    public Hola (int nivel) {
```

```
        this.nombre = "nn";  
        this.nivel = nivel;
```

```
    }
```

```
    public Hola (Hola H) {
```

```
        this.nombre = H.nombre;  
        this.nivel = H.nivel;
```

```
    }
```

```
    abstract void golpea();
```

```
    }
```

```
abstract class mesa {
```

```
    String nombre;  
    int nivel;
```

```
    public mesa () {}
```

```
    public mesa (String nombre, int nivel) {
```

```
        this.nombre = nombre;  
        this.nivel = nivel;
```

```
    public Hola (int nivel) {
```

```
        this.nombre = "nn";  
        this.nivel = nivel;
```

```
    }
```

```
    public mesa (mesa s) {
```

```
        this.nombre = s.nombre;  
        this.nivel = s.nivel;
```

```
    }
```

```
    abstract void Hola();
```

```
    }
```



```

abstract class Silla {
    String nombre;
    int nivel;

    public Silla () {}

    public Silla (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public Silla (int nivel) {
        this.nombre = "nn";
        this.nivel = nivel;
    }

    public Silla (Silla s) {
        this.nombre = s.nombre;
        this.nivel = s.nivel;
    }

    abstract void tola ();
}

```

```

abstract class Plato {
    String nombre;
    int nivel;

    public Plato () {}

    public Plato (String nombre, int nivel) {
        this.nombre = nombre;
        this.nivel = nivel;
    }

    public Plato (int nivel) {
        this.nombre = "nn";
        this.nivel = nivel;
    }

    public Plato (Plato p) {
        this.nombre = p.nombre;
        this.nivel = p.nivel;
    }

    abstract void plato ();
}

```


Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
export class Jug {
```

```
  jugar() {
```

```
    throw new Error("metodo jugar()");
```

```
  }
```

```
export class Ata {
```

```
  Atacar() {
```

```
    throw new Error("metodo atacar()");
```

```
  }
```

```
export class Def {
```

```
  Defec() {
```

```
    throw new Error("metodo defender()");
```

```
  }
```

```
export class Cur {
```

```
  cura() {
```

```
    throw new Error("metodo curar()");
```

```
  }
```

```
export class Mag {
```

```
  hechizo() {
```

```
    throw new Error("metodo hechizo()");
```

```
  }
```



```
export class Jugu4 {
```

```
  constructor ( nombre = "Steven", nivel = 1, copia = null ) {  
    if ( copia ) {  
      this.nombre = copia.nombre;  
      this.nivel = copia.nivel;  
    } else {  
      this.nombre = nombre;  
      this.nivel = nivel;  
    }  
  }  
  defender () {  
    throw new Error (" metodo defender ");  
  }  
}
```

```
export class Jugar {
```

```
  constructor ( nombre = "Steven", nivel = 3, copia = null ) {  
    if ( copia ) {  
      this.nombre = copia.nombre;  
      this.nivel = copia.nivel;  
    } else {  
      this.nombre = nombre;  
      this.nivel = nivel;  
    }  
  }  
  hechizo () {  
    throw new Error (" metodo hechizo () ");  
  }  
}
```

```
export class Jugar {
```

```
  constructor ( nombre = "Steven", nivel = 6, copia = null ) {  
    if ( copia ) {  
      this.nombre = copia.nombre;  
      this.nivel = copia.nivel;  
    } else {  
      this.nombre = nombre;  
      this.nivel = nivel;  
    }  
  }  
}
```

```
export class Jugar {
```

```
  constructor ( nombre = "Steven", nivel = 7, copia = null ) {  
    if ( copia ) {  
      this.nombre = copia.nombre;  
      this.nivel = copia.nivel;  
    } else {  
      this.nombre = nombre;  
      this.nivel = nivel;  
    }  
  }  
}
```



```

export class Joga {
  constructor ( nome = "Steven", nivel = 1, copia = null ) {
    if ( copia ) {
      this.nome = copia.nome;
      this.nivel = copia.nivel;
    } else {
      this.nome = nome;
      this.nivel = nivel;
    }
  }
  golp () {
    throw new Error (" metodo golpear() ");
  }
}

```

```

export class Jugar {
  constructor ( nome = "Steven", nivel = 2, copia = null ) {
    if ( copia ) {
      this.nome = copia.nome;
      this.nivel = copia.nivel;
    } else {
      this.nome = nome;
      this.nivel = nivel;
    }
  }
  ataca () {
    throw new Error (" metodo atacar() ");
  }
}

```

```

export class Jugar2 {
  constructor ( nome = "curador", nivel = 1, copia = null ) {
    if ( copia ) {
      this.nome = copia.nome;
      this.nivel = copia.nivel;
    } else {
      this.nome = nome;
      this.nivel = nivel;
    }
  }
  curar () {
    throw new Error (" metodo curar() ");
  }
}

```



```

export class Juguete {
  constructor (nombre = "Steven", nivel = 8, copia = null) {
    if (copia) {
      this.nombre = copia.nombre;
      this.nivel = copia.nivel;
    } else {
      this.nombre = nombre;
      this.nivel = nivel;
    }
  }
}

```

```

export class Juguete {
  constructor (nombre = "Steven", nivel = 9, copia = null) {
    if (copia) {
      this.nombre = copia.nombre;
      this.nivel = copia.nivel;
    } else {
      this.nombre = nombre;
      this.nivel = nivel;
    }
  }
}

```

```

export class Juguete {
  constructor (nombre = "Steven", nivel = 10, copia = null) {
    if (copia) {
      this.nombre = copia.nombre;
      this.nivel = copia.nivel;
    } else {
      this.nombre = nombre;
      this.nivel = nivel;
    }
  }
}

```


Nombre:

Fecha:

Profesor:

Materia:

Institución:

Curso:

Nota:

```
namespace Persona
{
    public interface Interfa
    {
        void Metodo();
    }
}
```

```
namespace mesa
{
    public interface interfa
    {
        void metodo();
    }
}
```

```
namespace PC
{
    public interface interfaz
    {
        void metodo();
    }
}
```

```
namespace silla
{
    public interface interfaz
    {
        void metodo();
    }
}
```

```
namespace lapiz
{
    public interface interfaz
    {
        void metodo ();
    }
}
```