Argentina programa 4.0

UNIVERSIDAD TECNOLOGICA NACIONAL

Repaso de Java

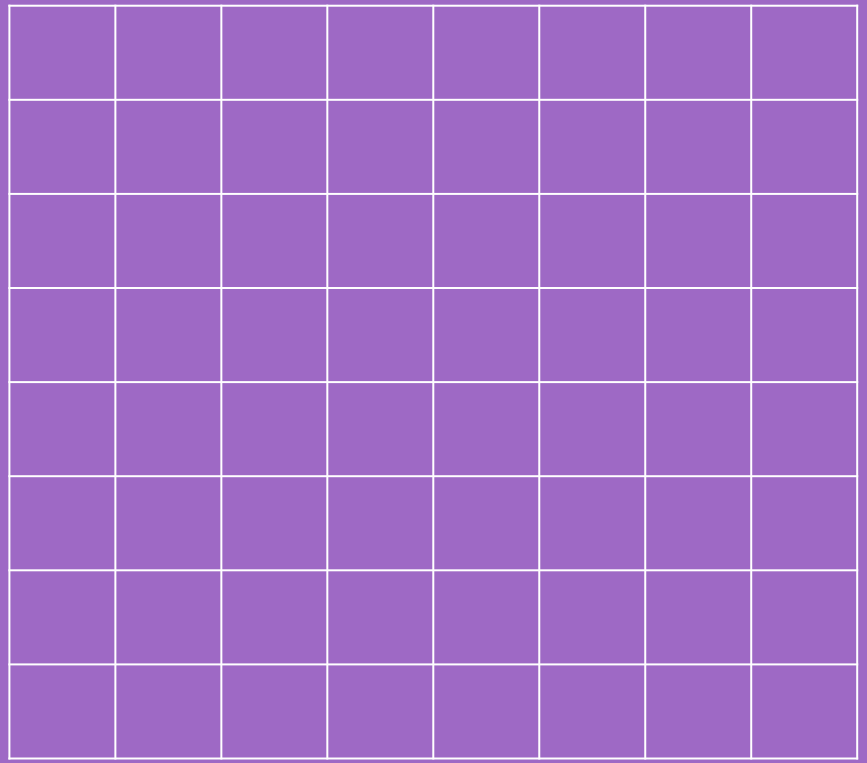
Tramo 2 - Desarrollador Java Intermedio

Agenda

Argentina programa 4.0

- Clasificación del lenguaje
- Sintaxis básica
- Método main
- Paradigma Orientado a Objetos
- Colecciones

Clasificación del lenguaje

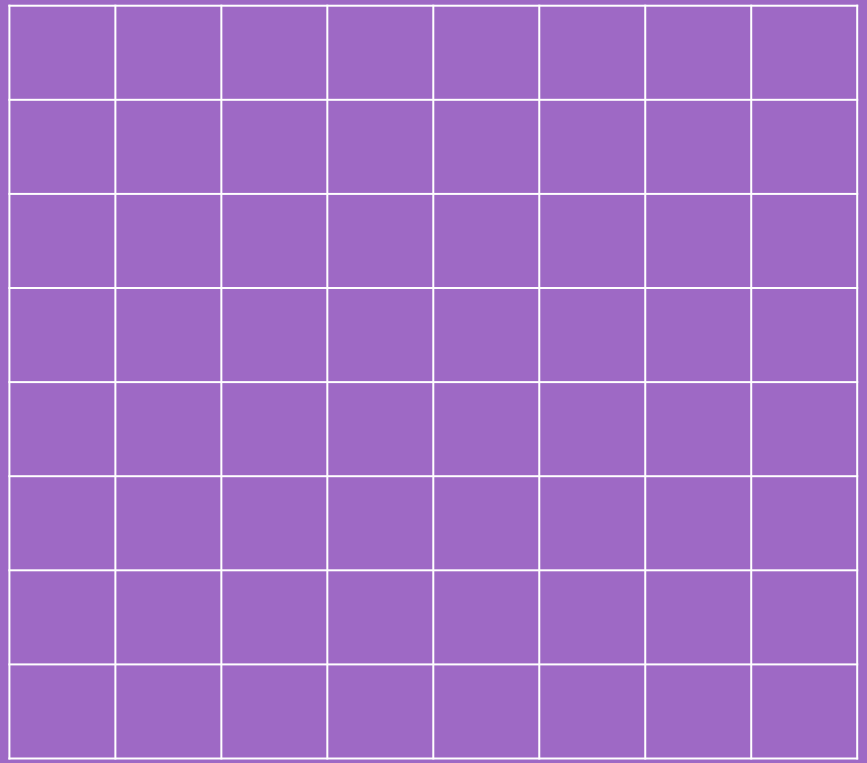


Clasificaciones

Argentina programa 4.0

- **Compilado**:
  - Se debe compilar antes de ejecutar
  - Código fuente se compila en bytecode que luego es ejecutado por una JVM
- **Fuertemente tipado**:
  - Toda variable/constante debe ser de un tipo de dato definido
  - No se pueden mezclar diferentes tipos de datos
- **Tipado estático**:
  - Validaciones de tipos realizadas en tiempo de compilación y no ejecución

Sintaxis básica



Variables y Tipos de datos primitivos

Argentina programa 4.0

```
char unaLetra = 'a';
boolean unValorBooleano = true;
int miPrimerContador = 66;
double unValor = 1.68;
float otroNum = 2.344f;
```

- No se le pueden invocar métodos (no tienen)
- Siempre tienen un valor NO nulo

Operadores y Expresiones

Argentina programa 4.0

Op Binarias básicas

```
10 + 20
15 - 12
10 * 3
8 / 3
8 % 3   // 2
2 ** 3  // 8
```

Precedencia

```
3 * 2 +  3

(3 * 2) +  3
```

```
int miPrimerContador = 66;
double unValor = 1.68;

miPrimerContador + 20
15 - 12
10 * 3
unValor / 3
8 % 3
```

Booleanos

Argentina programa 4.0

Operadores de comparación y predicados

```
10 > 20
15 >= 12
10 == 3
8 != 3
```

```
int miPrimerContador = 66;
double unValor = 1.68;
double otroValor = 1.67;

unValor == (otroValor + 0.01)
```

```
boolean unBooleano = true;
boolean otroBooleano = false;

! unBooleano // false
unBooleano && otroBooleano // false
unBooleano || otroBooleano // true

unBooleano && (otroBooleano || True) // true
```

Condicionales

Argentina programa 4.0

```
if(unValor < otroNum) {
      //una acción
   }
   if(unValor < otroNum) {
      //una acción
   } else {
     //otra acción
   }
```

```
char unaLetra = 'a';
   switch (unaLetra){
     case 'b':
       //Hacer A
       break;
     case 'a':
       //Hacer B
       break;
     default:
       //Hacer Z
   }
```

```
unBooleano && (otroBooleano || True)
if(unValor < otroNum) {
     //una acción
   }

int unValor = 1;
int otroValor = 2;

boolean unaCond = unValor == (otroValor + 1);

if(unaCond) {
     //hacer algo
}
```

Bucles

Argentina programa 4.0

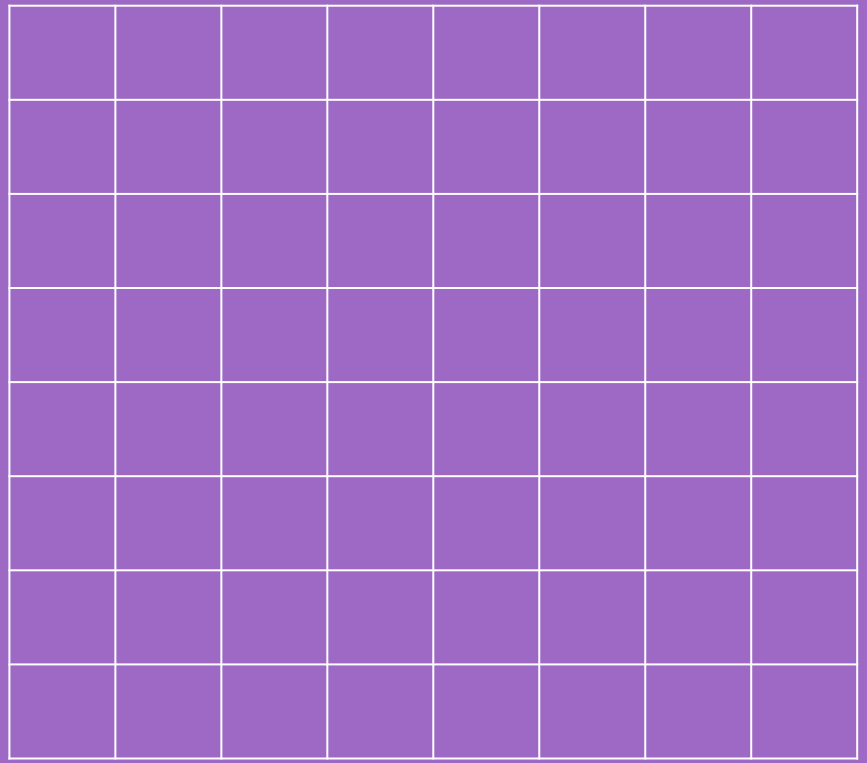
```
while(condicionX){
     //Una Acción
     //En algún momento se tiene
     // modificar la condicionX
   }

for(inicia;condicionX;cambiaElemento){
   //Una Acción
}
```

```
int unNum = 10;
while(unNum > 0){
     System.out.println(unNum);
     unNum = unNum -1;
   }

   for(int otroNum=0;otroNum<10;otroNum++){
     System.out.println(otroNum);
   }
```

Método main



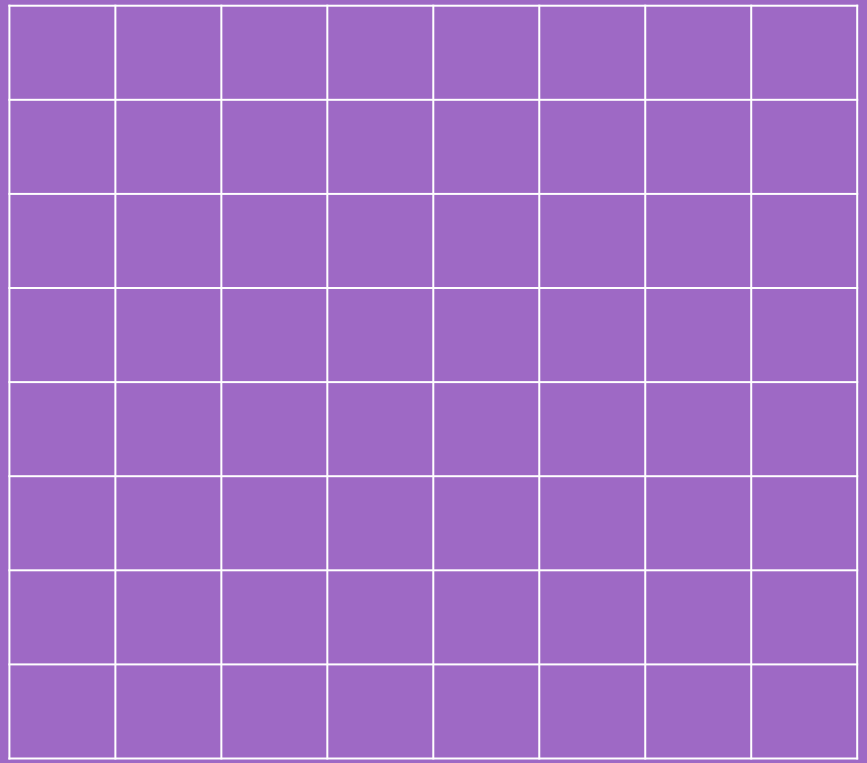
Firma y características

Argentina programa 4.0

```
public static void main(String[] args)
{
    ...
}
```

- Obligatorio en toda aplicación Java
- Punto de entrada de la aplicación

Paradigma Orientado a Objetos



Clase

```
public class UnaClase {
	private String unAtributo;

	//Constructor
	public UnaClase(String unAtributo) {
		this.unAtributo = unAtributo;
	}

	//Un método
	public String getUnAtributo() {
		return this.unAtributo;
	}
}
```

- Comportamiento y atributos (no valores) compartidos por un conjunto de objetos. Los atributos pueden ser de algún tipo de dato primitivo o de alguna otra clase.
- Elementos de Instancia:
  - Variables: tienen un valor distinto por Objeto
  - Métodos: código a ejecutar, tiene acceso a variables de instancia
- Cuentan con un constructor vacío por default. En caso de definir un constructor no vacío, como en el ejemplo, si se desea mantener el constructor vacío, el mismo debe ser declarado.

Objeto

Argentina programa 4.0

```
UnaClase unObjeto = new UnaClase("valorUnAtributo");
```

- Un objeto es una instancia de una determinada Clase
- Todo objeto se crea a través de un constructor. Con Java, el mismo se llama utilizando la palabra reservada new
- Para utilizar un método de un objeto, decimos que al mismo se le envía un “mensaje”. Para representar esto, se pone un punto “.” luego de una variable referenciando un objeto:

```
unObjeto.getUnAtributo()
```

Clase abstracta

- Se declaran con “abstract”
- Puede tener métodos abstractos y no abstractos
- No se puede instanciar
- Una clase, en Java, puede extender directamente de una única clase abstracta

```
public abstract class Descuento {
	private float valorDescuento;

	public float getValorDescuento() {
		return this.valorDescuento;
	}

	public void setValorDescuento(float valorDescuento) {
		this.valorDescuento = valorDescuento;
	}
}
------------------------
public class DescuentoFijo extends Descuento {
	@Override
	public float valorFinal(float valorInicial) {
		return valorInicial - this.getValorDesc();
	}
}
--------------------
public class DescuentoPorcentaje extends Descuento {
	@Override
	public float valorFinal(float valorInicial) {
		return valorInicial - (valorInicial * this.getValorDesc());
	}
}
```

Argentina programa 4.0

Interfaces

- Se declaran con “interface”
- Contrato a cumplir
- Sin estado (no lleva atributos)
- Una clase puede implementar múltiples
- Se pueden definir constantes
- Semánticamente != heredar

```
public interface UnaInterface {

      public void hacerAlgo(Producto p);

      public Producto obtenerProducto(String nombre);

}
```

```
public class ProcesadorProducto implements UnaInterface {

      @Override
      public void hacerAlgo(Producto p) {
            //hacer algo….
      }

      @Override
      public Producto obtenerProducto(String nombre) {
            //retornar algo
            return algo;
      }
}
```

Enums

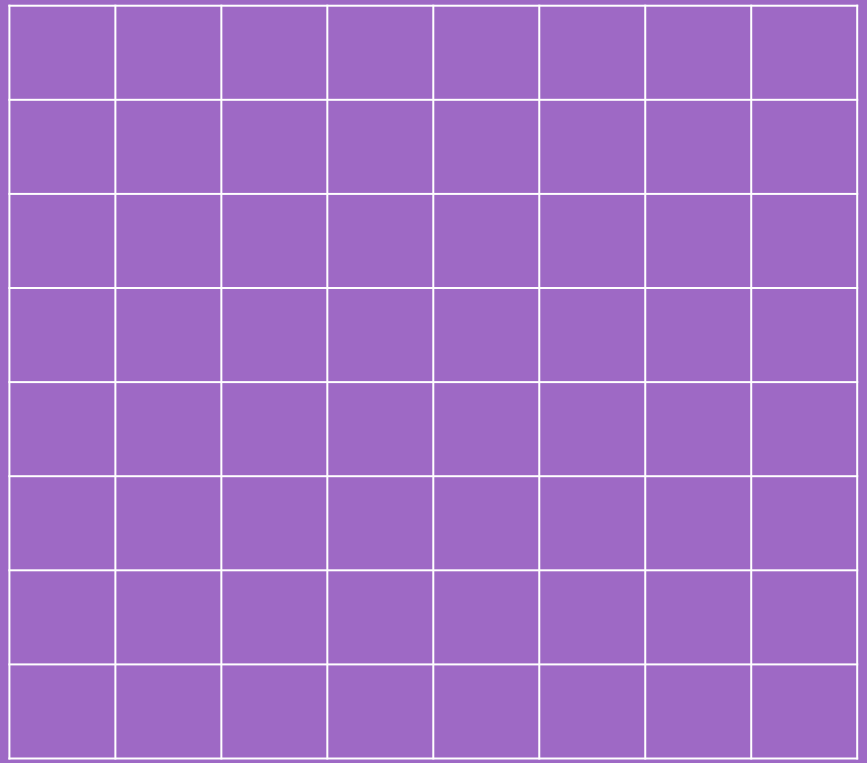
Argentina programa 4.0

```
public enum Dia {
    LUNES, MARTES, MIERCOLES, JUEVES,VIERNES, SABADO, DOMINGO
}
```

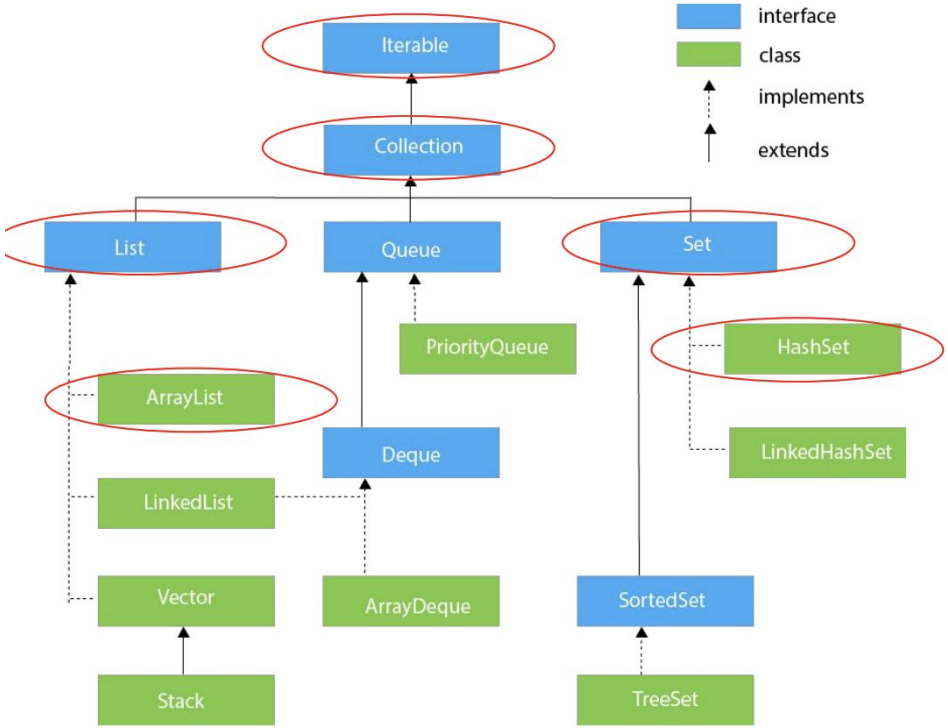
- Tipo de dato especial
- Representan constantes predefinidas
- Se acceden:

```
Dia.LUNES
```

Colecciones

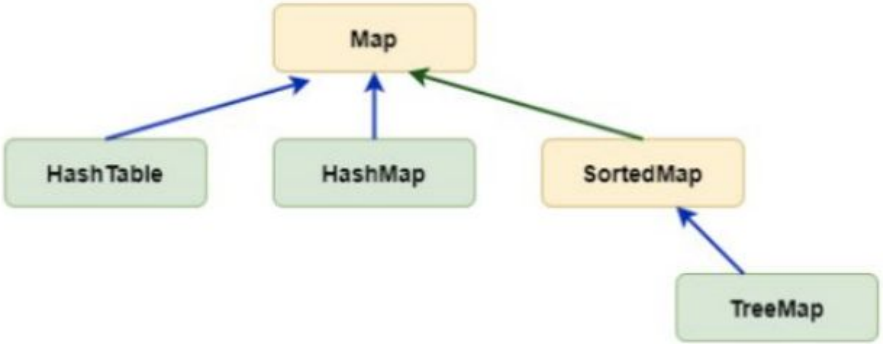


Colecciones



- Iterable -> hasNext / next
- Collection -> add / remove / contains
- List -> add(int index, E element) / get (int index)
- Set → no permite repetidos
- Varían en la representación interna de los datos, por ende van tener distintos tiempos de acceso, lectura , escritura según los datos que tengan.

Mapas



- put(key,value)
- get (key) : value
- hasKey : boolean
- keySet : Set

Gracias!