POLITÉCNICO
ESTELLA

ASIGNATURA: ETORNOS DE DESARROLLO

GRADO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

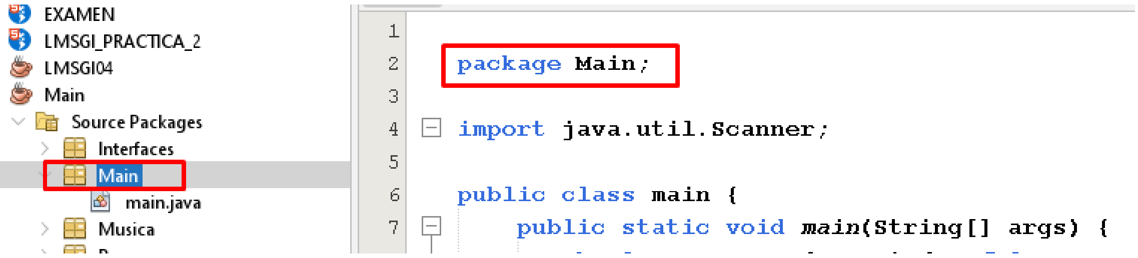
TAREA EDDE 04

Presentado por: Eugen Moga

Indice

Ejercicios 1

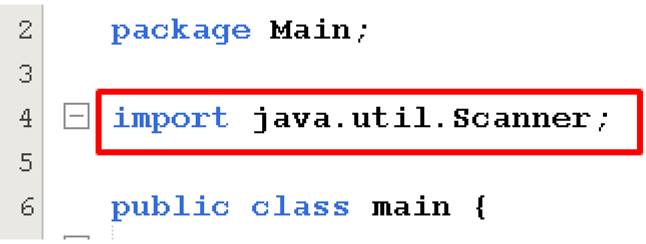
- Paquete



```

package Main;

import java.util.Scanner;

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

package Main; hace referencia al paquete

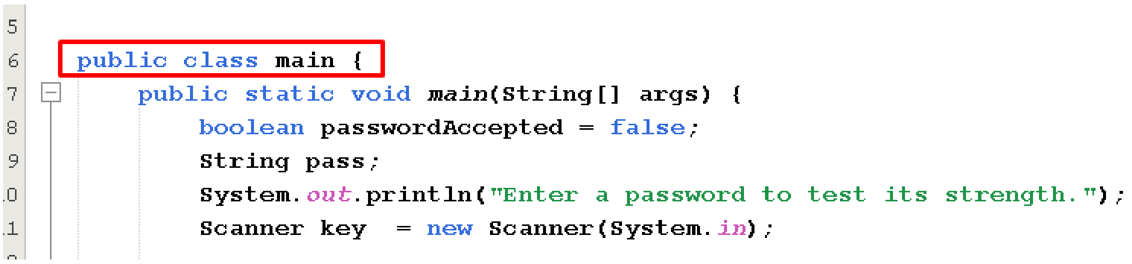
- Librerías/bibliotecas



```
package Main;

import java.util.Scanner;

public class main {
```

Con import se importa la clase Scanner de la biblioteca de Java para leer la entrada de usuario. En la librería de java hay muchas clases disponibles que hay que importar antes de poder utilizar

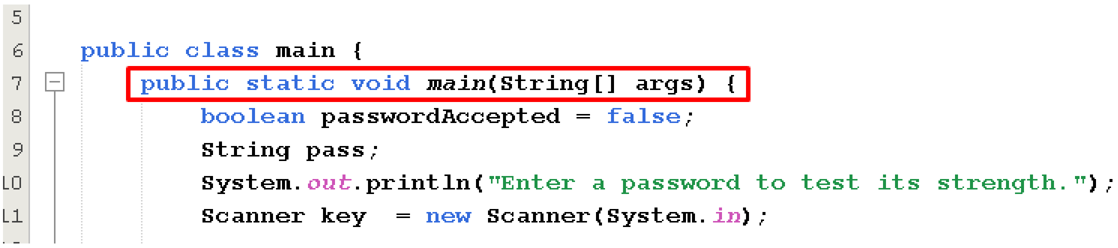
- Clase principal



```

public class main {
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key  = new Scanner(System.in);
```

Nombre de la clase principal, en este caso se llaman igual que el método main.

- Métodos/funciones



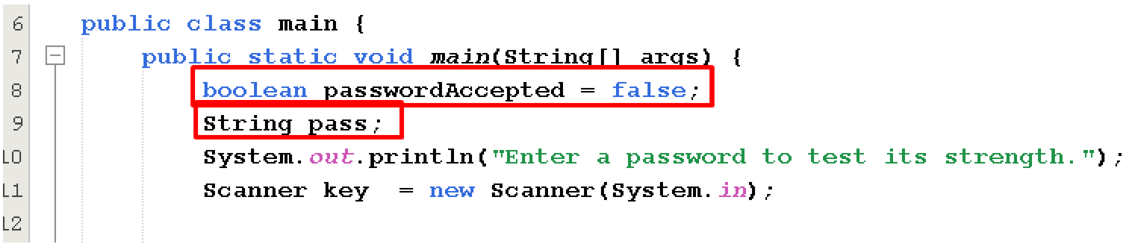
```
public class main {
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key  = new Scanner(System.in);
```

public static void main es la clase principal y es por donde se empieza a ejecutar el programa.

```
    public static boolean passwordIsGood(String pass){
        if(length(pass) && containsDigits(pass) &&
                containsUpLowCase(pass) && hasSpecial(pass)){
            System.out.println("Your password is strong!");
            return true;
        }
        System.out.println("Weak password.  Try another.");
        return false;
    }

    public static boolean length(String pass){
        if(pass.length() > 6){
            return true;
        }
        System.out.println("You need at least 6 characters.");
        return false;
    }

    public static boolean containsDigits(String pass){
        if(!pass.matches(".*\\d.*")){
            return true;
        }
        System.out.println("You need a digit.");
        return false;
    }

    public static boolean containsUpLowCase(String pass){
        if(!pass.equals(pass.toLowerCase()) && !pass.equals(pass.toUpperCase())){
            return true;
```

Metodos de la clase con distintas funcionalidades de comprobación y verificación para comprobar una contraseña

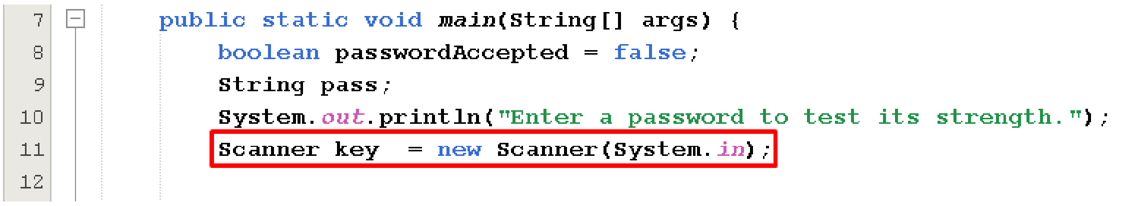
- Variables



```
public class main {
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key  = new Scanner(System.in);
```

Variable para almacenar datos de tipo String y boleano.

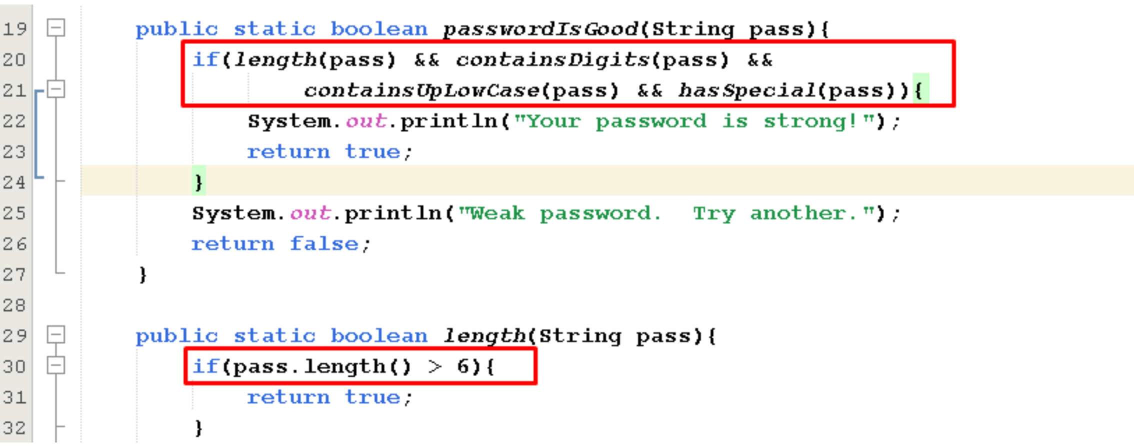
- Instanciaciones



```
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key  = new Scanner(System.in);

```

Se instancia un objeto de la clase Scanner

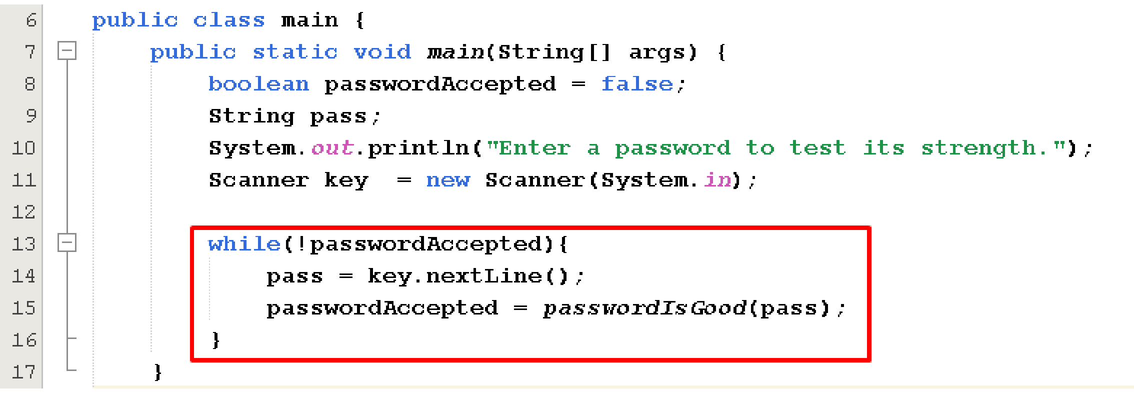
- Condicionales



```
    public static boolean passwordIsGood(String pass){
        if(length(pass) && containsDigits(pass) &&
                containsUpLowCase(pass) && hasSpecial(pass)){
            System.out.println("Your password is strong!");
            return true;
        }
        System.out.println("Weak password.  Try another.");
        return false;
    }

    public static boolean length(String pass){
        if(pass.length() > 6){
            return true;
        }
```

Se usa el if como condicional para verificar si la contraseña cumple con los criterios especificados.

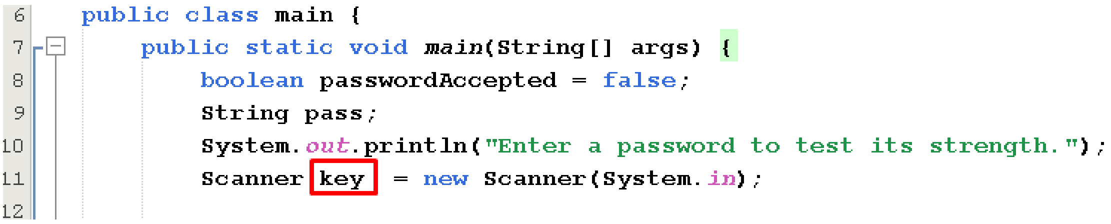
- Bucles



```
public class main {
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key  = new Scanner(System.in);

        while(!passwordAccepted){
            pass = key.nextLine();
            passwordAccepted = passwordIsGood(pass);
        }
    }
```

El bucle while en este caso se utiliza para pedir una contraseña hasta que se ingresa una contraseña valida por los requisitos.

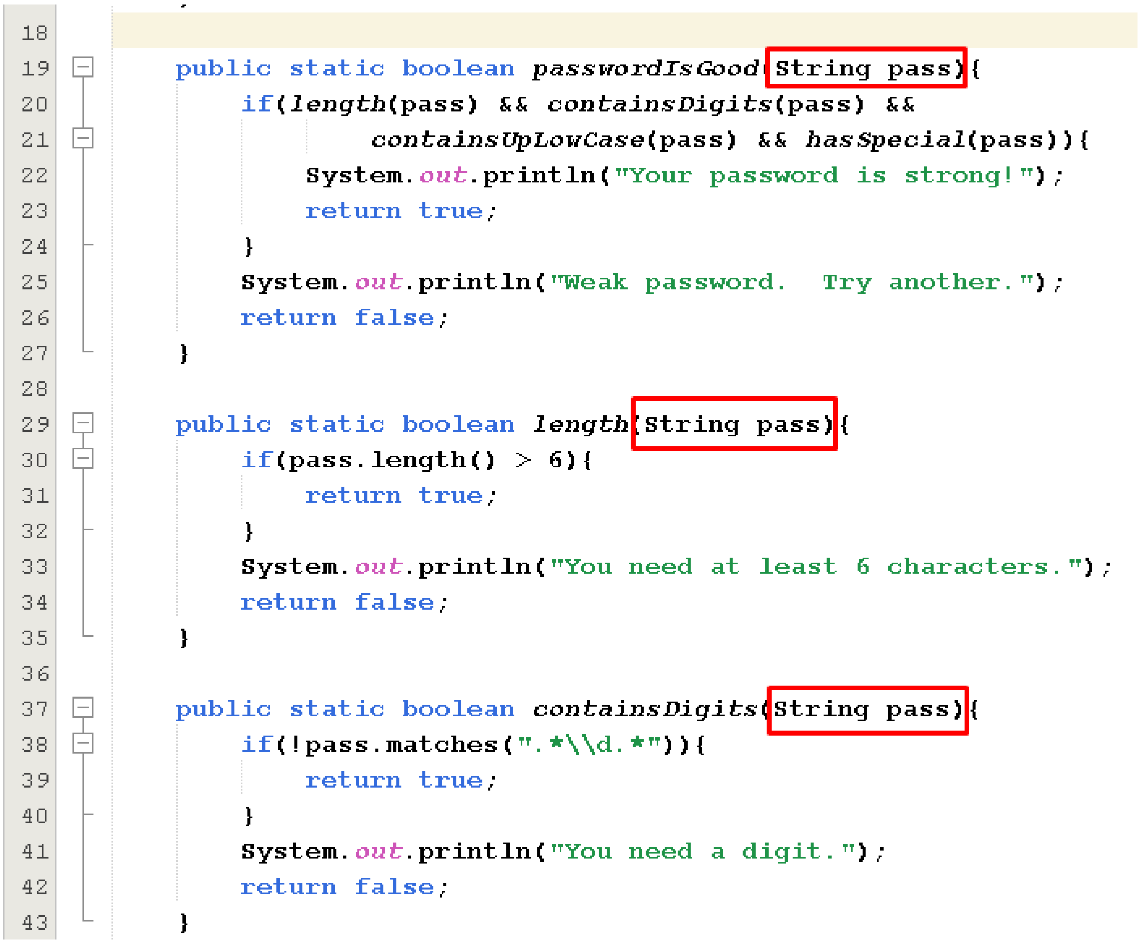
- Objeto



```java
public class main {
    public static void main(String[] args) {
        boolean passwordAccepted = false;
        String pass;
        System.out.println("Enter a password to test its strength.");
        Scanner key = new Scanner(System.in);

```

En este caso key es un objeto de la clase Scanner.

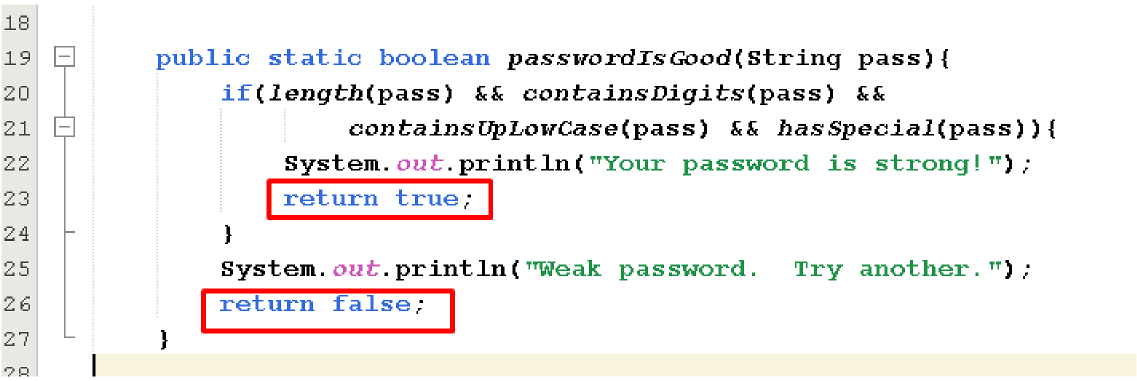
- Parámetros/argumentos (de un método)



```java

    public static boolean passwordIsGood(String pass){
        if(length(pass) && containsDigits(pass) &&
                containsUpLowCase(pass) && hasSpecial(pass)){
            System.out.println("Your password is strong!");
            return true;
        }
        System.out.println("Weak password.  Try another.");
        return false;
    }

    public static boolean length(String pass){
        if(pass.length() > 6){
            return true;
        }
        System.out.println("You need at least 6 characters.");
        return false;
    }

    public static boolean containsDigits(String pass){
        if(!pass.matches(".*\\d.*")){
            return true;
        }
        System.out.println("You need a digit.");
        return false;
    }
```

La variable pass es pasado como parámetro en los diferentes métodos.

- Valor de retorno (de un método)



```
    public static boolean passwordIsGood(String pass){
        if(length(pass) && containsDigits(pass) &&
                containsUpLowCase(pass) && hasSpecial(pass)){
            System.out.println("Your password is strong!");
            return true;
        }
        System.out.println("Weak password.  Try another.");
        return false;
    }
```

Los métodos de este programa validan si se cumplen las condiciones, si se cumple retorna verdadero y si no se cumple retorna falso.

Ejercicios 2

¿Cuáles son las ventajas de las herramientas de control de versiones de un proyecto?

Mi respuesta a la argumentación de Luis:

Hola Luis, solo tenias que contestar a una pregunta.

Estoy de acuerdo con tu respuesta sobre las ventajas de las herramientas de control de versiones de un proyecto, pero quiero añadir que Git también mejora la calidad del código mediante revisiones y “pull requests”

Esto de pull requests básicamente es que el propietario de un fork (rama) hace una petición al propietario del repositorio original para que este ultimo incorpore los commits que están en el fork. Y el propietario principal del proyecto revisa el código antes de incorporarlo al proyecto principal.

Para más información revisar este enlace.

Un saludo,
Eugen