

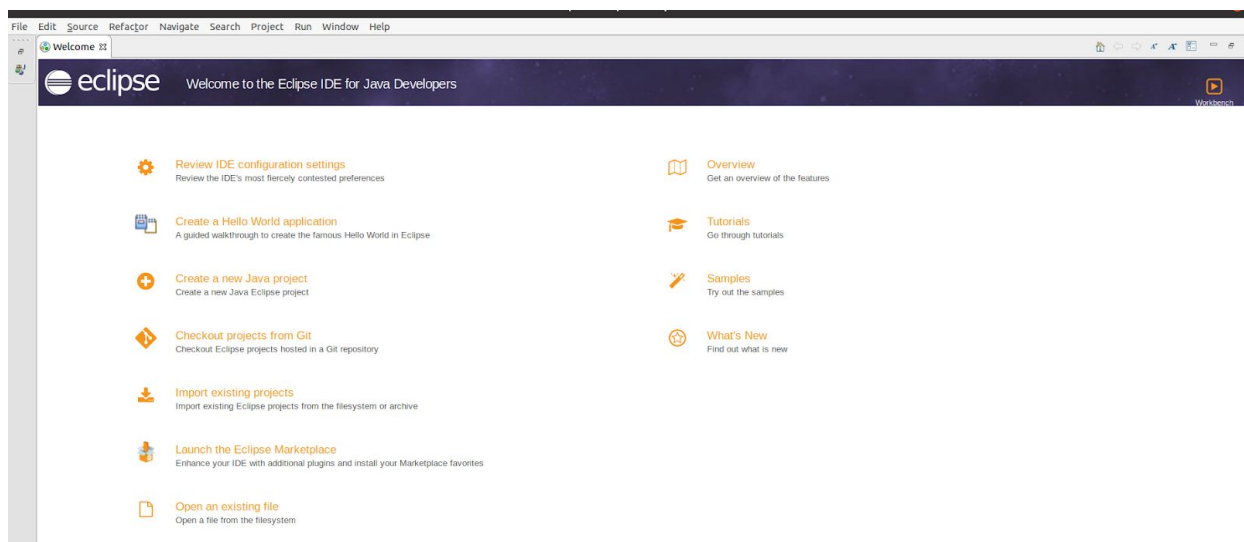
Pràctica 1: Ubicació de la programació

L'objectiu d'aquesta pràctica és fer una introducció a la programació.

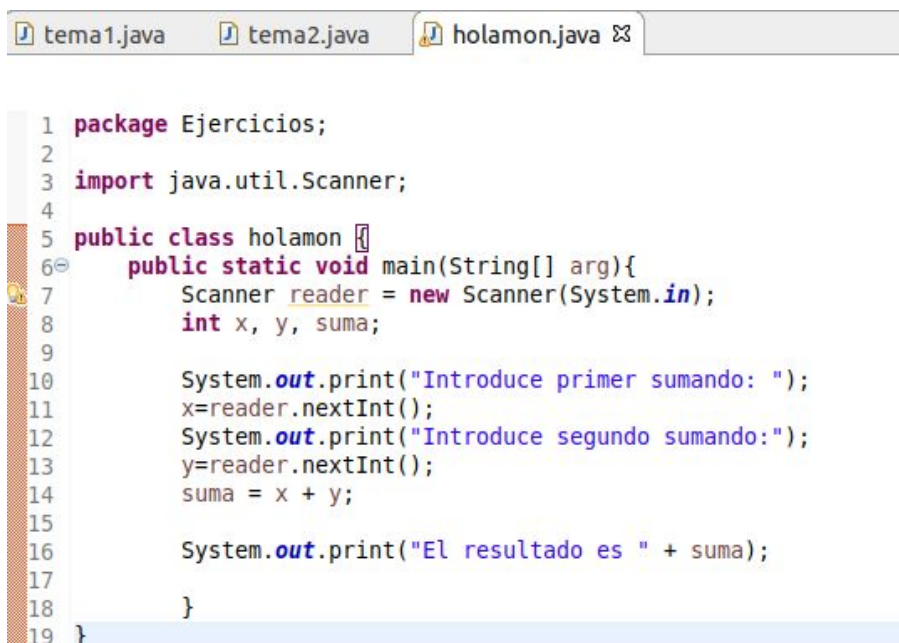
Part 1

Realitza les següents instruccions, fent les captures de pantalla dels processos involucrats o de la seva finalització segons es demani:

1. Realitza la Instal·lació del programari Eclipse (en windows/linux).



2. Crea un projecte nou de java anomenat holamon i copia el següent codi en un fitxer holamon.java.



Entorns de desenvolupament.

Desenvolupament de programari - UF1

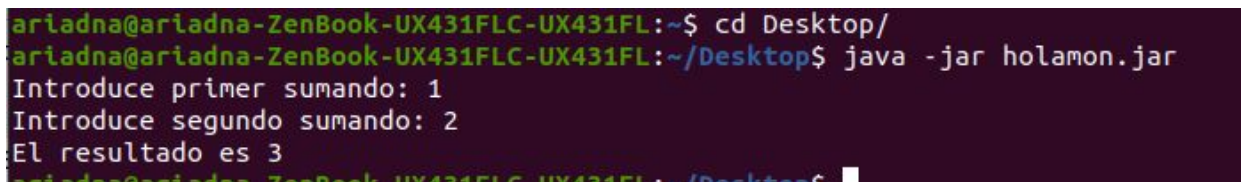
3. Executa el programa, comprova el seu funcionament. Busca el fitxer intermig que s'ha generat.



```
<terminated> holamon [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (Oct 9, 2020, 7:59:24 PM)
Introduce primer sumando: 4
Introduce segundo sumando: 3
El resultado es 7
```

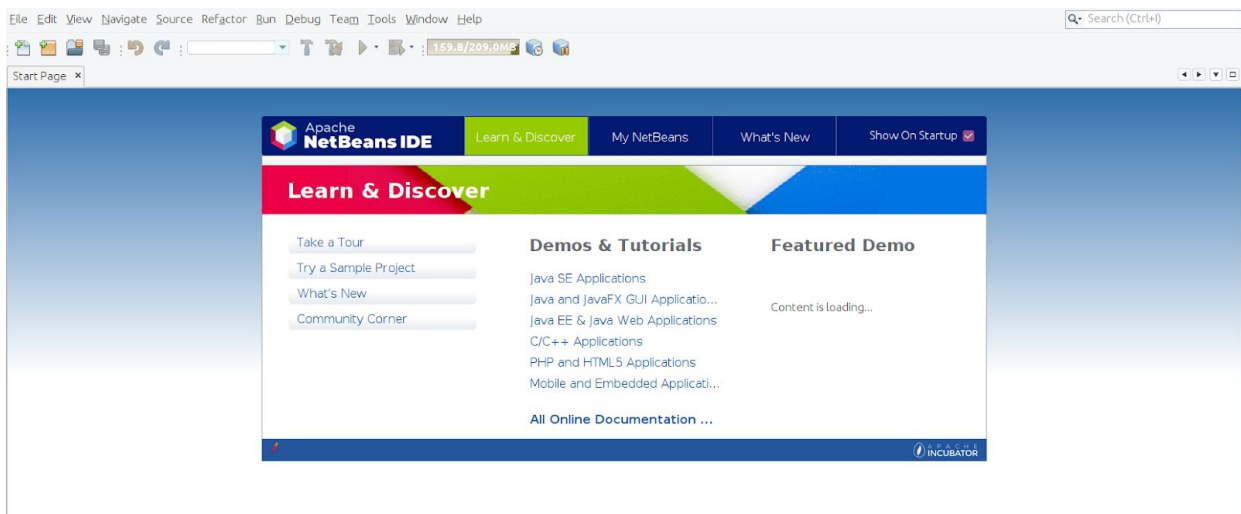


4. Obté un executable del programa realitzat, i executa'l mitjançant comandes de terminal.



```
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~$ cd Desktop/
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~/Desktop$ java -jar holamon.jar
Introduce primer sumando: 1
Introduce segundo sumando: 2
El resultado es 3
```

5. Realitza la instal·lació del programari Netbeans o IntelliJ idea.



Desenvolupament de programari - UF1

6. Crea un projecte nou de java anomenat holamon i copia el següent codi en un fitxer holamon.java.

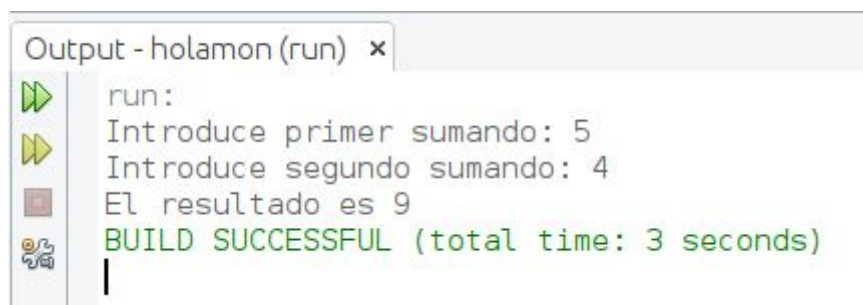
```
/**
 *
 * @author ariadna
 */
public class Holamon {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner reader = new Scanner(System.in);
        int x, y, suma;

        System.out.print("Introduce primer sumando: ");
        x=reader.nextInt();
        System.out.print("Introduce segundo sumando: ");
        y=reader.nextInt();
        suma = x + y;
        System.out.println("El resultado es " + suma);
    }

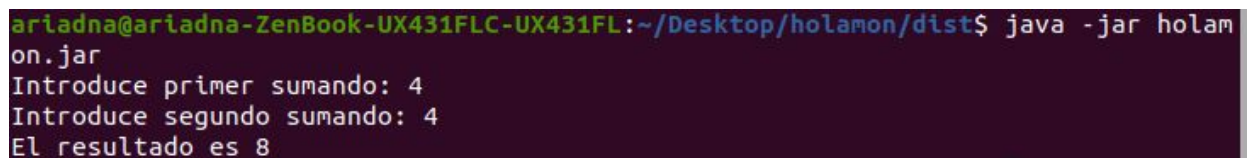
    // TODO code application logic here
}
```

7. Executa el programa, comprova el seu funcionament. Busca el fitxer intermig que s'ha generat.



```
Output - holamon (run) x
run:
Introduce primer sumando: 5
Introduce segundo sumando: 4
El resultado es 9
BUILD SUCCESSFUL (total time: 3 seconds)
```

8. Obté un executable del programa realitzat, i executa'l mitjançant comandes de terminal.



```
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~/Desktop/holamon/dist$ java -jar holamon.jar
Introduce primer sumando: 4
Introduce segundo sumando: 4
El resultado es 8
```

Part 2

Realitza les següents instruccions, fent les captures de pantalla dels processos involucrats o de la seva finalització segons es demani:

```
#include <stdio.h>
int main() {
    int a, b;
    printf("Introduce numero 1:");
    scanf("%d",&a);
    printf("Introduce numero 2:");
    scanf("%d", &b);
    int z = a + b;
    printf("El resultado es %d",z);
    scanf("%d", &b);
}
```

1. Copia el següent codi en un fitxer holamon.c. NOTA: sudo nano holamon.c (en linux)

```
#include <stdio.h>
int main() {
    int a, b;
    printf("Introduce numero 1:");
    scanf_s("%d",&a);
    printf("Introduce numero 2:");
    scanf_s("%d", &b);
    int z = a + b;
    printf("El resultado es %d",z);
}
```

2. Obté un executable del programa realitzat. Quin programari realitza aquesta funció. Indica la comanda que transforma holamon.c en un executable.



El programari que realitza el executable es el gcc. La comanda que transforma en un executable es: gcc holamon.c -o holamon. I la comanda per executar es: ./holamon

NOTA: L'opció més adequada en windows per executar-ho és Visual Studio Professional. En Linux l'habitual es fer-ho des de la línia de comandes amb el programari **gcc holamon.c -o holamon**

3. Executa el programa.

```
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~$ ./holamon
Introduce numero 1: 45
Introduce numero 2: 46
El resultado es 91ariadna@ariadna-ZenBook-UX431FLC-UX4
```

Desenvolupament de programari - UF1

Part 3

Realitza les següents instruccions, fent les captures de pantalla dels processos involucrats o de la seva finalització segons es demani:

1. Realitza la Instal·lació del programari Python <https://www.python.org>.

```
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~$ ls /usr/bin/python*  
/usr/bin/python2      /usr/bin/python3-futurize  
/usr/bin/python2.7    /usr/bin/python3-pasteurize  
/usr/bin/python3      /usr/bin/python-argcomplete-check-easy-install-script3  
/usr/bin/python3.8    /usr/bin/python-argcomplete-tcsh3
```

2. Copia el següent codi en un fitxer holamon.py. NOTA: sudo nano holamon.py (en linux)

```
def p2():  
    x=int(input('Introduce numero 1: '))  
    y=int(input('Introduce numero 2: '))  
    z=x+y  
    print "Resultado:" + str(z)  
  
if __name__ == "__main__":  
    p2()
```

3. Executa el programa amb l'interpret de python.

```
ariadna@ariadna-ZenBook-UX431FLC-UX431FL:~$ python3 holamon.py  
Introduce numero 1: 12  
Introduce numero 2: 23  
Resultado:35
```

Part 4

Seguiu els passos de la activitat per tal de generar els següents fitxers de codi: holamon, holamon.exe, holamon.c, holamon.class, holamon.java i holamon.py.

Contesta a les següents qüestions:

4. Dels fitxers de codi anteriors, identifica quins son de codi font i quins de codi objecte. Hi ha algú que no correspongui a cap de les categories anteriors.

Holamon és codi objecte

holamon.exe és codi objecte

holamon.c és codi objecte

holamon.class és font objecte

holamon.java és codi font

holamon.py fa les dues coses

Desenvolupament de programari - UF1

5. Dels fitxers anterior, quins podem executar directament, quins invocant un intèrpret i quins no podem executar directament.

Holamon es pot executar directament però no en un Windows, només en Linux.

holamon.exe es pot executar directament a Windows, però no a Linux.

holamon.c no es pot executar directament

holamon.class sí es pot executar directament

holamon.java no es pot executar directament

holamon.py es sí es pot executar directament

Part 5

Imagina't que copiem els fitxers anteriors en un entorn Windows ®. Contesta **raonadament** a les següents qüestions:

1. Es podria executar holamon en Windows? Què caldria fer per executar holamon.c en windows?

No es podria executar holamon en Windows perquè es un programa compilat per al sistema Linux. Per executar holamon.c a Windows s'hauria de a compilar tot en un visual Studio, o un altre programa, que compiles en un sistema Windows.

2. Es podria executar holamon.py en Windows? Què caldria fer per executar holamon.py en windows?

Sí que es podria executar holamon.py al sistema Windows.

3. Es podria executar holamon.class en Windows? I holamon.java Què caldria fer per executar holamon.class en windows?

Sí que es podria executar holamon.java al Windows. El holamon.java no es podria per què no esta compilat, faltaria compilar-lo per fer-lo executar a Windows.

Part 6

Contesta raonadament les següents qüestions:

1. Explica detalladament les diferències en el funcionament entre un llenguatge interpretat, un compilat i un amb màquina virtual.

El **llenguatge compilat** consta de diferents processos que consisteixen en verificar que tot estigui correctament escrit i fer una primera traducció a un codi intermig anomenat codi objecte.

El codi objecte resultants, se li afegixen altres codis objectes de llibreries del llenguatge i es fa la traducció final a llenguatge de màquina. Per tant, la traducció a llenguatge màquina es fa abans i l'execució del programa es fa després. Per cada sistema operatiu s'ha de realitzar una traducció

Entorns de desenvolupament.

Desenvolupament de programari - UF1

diferent. Al final del procés de traducció disposaren d'un arxiu que es podrà executar directament sobre el sistema operatiu. En canvi, el **llenguatge interpretat** no hi ha cap procés de traducció. El codi font l'interpreta un altre programa que va llegint línia a línia i les va executant. Si es vol tornar a executar, tornarà a interpretar cadascuna de les instruccions. Cada sistema operatiu pot tenir el seu propi interpret. Generalment son de web (javascript, php, asp, o com python, Perl,...). I per últim, el **llenguatge de màquina virtual**, es un híbrid, el codi font es compila a un codi intermig que és comú a tots els sistemes, i cada sistema operatiu disposa d'un software anomenat màquina virtual que s'encarrega d'agafar el codi intermig i fer la interpretació final.

2. Avantatges i inconvenients de cadascun (característiques com ara la velocitat d'execució, la facilitat/flexibilitat de codificació i depuració). Fes-ho en format de taula d'aquest tipus

	L. interpretat	L. compilat	màquina virtual
Velocitat d'execució	Lent	Ràpid	Entre mig del interpretat i el compilat
Procés de traducció	Cada vegada	Una sola vegada	Una traducció a codi intermig
Genera executable	No	Sí	Un executable portable entre plataformes
Sistemes operatius	Interprete per cada sistema	S'ha de adaptar al sistema	Màquina virtual
Complexitat	Menys flexibles y potents	Potents i flexibles	Dependents de la plataforma
Codi font	El codi font es necessari en cada execució	Una vegada compilat, el codi font pot quedar en secret	Només fa falta el codi intermig

Errors	Els errors sintàctics es detecten durant l'execució	Es detecten duran la compilació	Es detecten duran la compilació
--------	---	---------------------------------	---------------------------------

Part 7

Llegeix l'article de la wikipedia Llenguatge de programació o bé, descarregat la versió en pdf.. (Atenció: no importa si hi ha algunes (o moltes) coses que no entens d'aquest article. Algunes et semblaran més clares amb el temps. D'altres no i, probablement, tampoc no serà greu. Per tant, tranquil·litat).

Argumenta sobre les següents preguntes sobre l'article:

1. Quin programa que escriuen "hola" per pantalla creus que s'executarà més ràpidament, la versió en assembler de x86 o la versió python. Creus que pot valer la pena la diferència? Perquè?

La versió x86 perquè el llenguatge està escrit amb més baix nivell, més proper al binari, ha de fer menys traducció, llavors serà més ràpid que el python, ja que el python és més complexe, l'ordinador ha de traduir el llenguatge.

Crec que entre aquests dos llenguatges no val la pena, perquè el temps que perds en escriure en x86, no el recuperes amb la rapidesa d'execució. És millor un llenguatge que és més proper al humà.

2. Troba al menys un exemple de llenguatge per cadascun dels paradigmes indicats: imperatiu, funcional, lògic i orientat a objectes. Pots fer servir la llista de llenguatges que proporciona l'article. Afegeix un tros de codi per il·lustrar cada llenguatge.

Declaració imperativa en PHP:

```
1 $listaParticipantes = [1 => 'Peter', 2 => 'Hans', 3 => 'Sarah'];
2 $nombres = [];
3 foreach ($listaParticipantes as $id => $name) {
4     $nombres[] = $name;
5 }
```

Declaració declarativa: (PHP)

```
1 $nombres = array_values($listaParticipantes);
```


Desenvolupament de programari - UF1

Imperativa (java)

Declarativa

```
int suma = 0;
for (int i = 1 to 100)
    suma += i;
return suma;
```

```
suma = Sum(1, 100)
```

Orientada a objectes: (S'han de declarar classes) (Llenguatge C++)

```
class Aritmetica
{
    public:
        inline int sumar (int a, int b) const
        {
            return a + b;
        }

        inline int restar (int a, int b) const
        {
            return a - b;
        }

        inline float multiplicar (int a, int b) const
        {
            return a * b;
        }

        inline float dividir (int a, int b) const
        {
            return a / b;
        }
};
```

Programació lògica (permet crear hipotesis)

- *Hermano(Ricardo, Juan)*

Programació funcional (Scheme)

```
1  2 ; → 2
2  (+ 2 3) ; → 5
3  (+) ; → 0
4  (+ 2 4 5 6) ; → 17
5  (+ (* 2 3) (- 3 1)) ; → 8
```

3. D'entre les coses que no has entès de l'article, escull una que t'hagi cridat especialment l'atenció. Proposa alguna explicació que et sembli mínimament plausible sobre el seu significat.

Diferencia entre semàntica estàtica i dinàmica:

Partint de la base que la semàntica descriu el significat dels símbols, paraules i frases d'un llenguatge, ja sigui natural o informàtic. Llavors, la semàntica estàtica es tracta de comprovar en el moment de la compilació, i la semàntica dinàmica es la que comprova en el moment de l'execució- .