

Guida introduttiva all'utilizzo di JAVA



Indice

1	Utilizzo dell'IDE	1
2	Gestione di file condivisi	3
2.1	Introduzione	3
2.2	Prerequisiti	3
2.3	Download dei file in repository	5
2.4	Abilitare le Notifiche	6
2.5	Importare i progetti in Eclipse	7
2.6	Abilitare le librerie	8
2.7	Settare la Run Configuration	9
2.7.1	Settare variabili d'ambiente	10
2.7.2	Project e Main Class	11
2.7.3	Argomenti	11
2.8	Aggiornamenti di grandi moli di dati	11
3	Utilizzo di GIT	13
3.1	Alcuni comandi base	13
3.2	Push e Pull	13
4	Organizzazione del Lavoro: svolgimento dei test.	15
4.1	Introduzione	15
4.2	Come effettuare i test	15
4.3	Lettura di dati da un xml	16
4.3.1	Caratteri speciali in xml	17
4.4	Input da File	18
4.4.1	Scrivere e salvare il file	18
4.4.2	Leggere un file xml	19
4.5	Generare un default Aircraft	19

INTRODUZIONE

In questo fascicolo saranno trattati tutti gli step necessari per installare JAVA sul proprio pc e per poter effettuare la condivisione di file e progetti. Gli argomenti sono trattati sequenzialmente a partire da zero. Si farà riferimento al codice di ADOPT. Prima di tutto sarà necessario avere un ambiente di lavoro per JAVA (CAP 1). JAVA rappresenta un linguaggio di programmazione ed è pertanto possibile scrivere un programma JAVA su un qualsiasi editor di testo (Notepad ++, TextPad etc.) e poi fare il run dal prompt dei comandi. Questo approccio può andare bene per i programmi più semplici ma per lo sviluppo di codici più complessi viene fatto uso di ambienti di sviluppo integrati, i cosiddetti IDE. Un IDE è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente, in pratica è come un editor di testo ma con tutte le funzionalità per aiutare nella programmazione, come correggere errori nel codice, completare i comandi etc. Il capitolo 1 tratterà come è possibile scaricare un IDE.

In seguito, per poter lavorare su un lavoro in condivisione, sarà necessario avere una piattaforma che consenta la condivisione di progetti, anche grandi. A tal pro occorrerà avere un Account su Bitbucket.org ed essere invitati alle *Repository* (letteralmente depositi, cioè gli archivi) per poter operare sui file in essi. In pratica tutti i progetti condivisi verranno messi in Repository su Bitbucket e gli utenti potranno fare il download dei progetti presenti per visionarli e, se autorizzati, anche l'upload delle proprie modifiche.(CAP 2) (CAP 3).

Per fare ciò si utilizza un software di gestione di comandi, GIT (CAP 4), che presenta una bash, simile al prompt dei comandi di windows, tramite la quale si può operare sui file in repository.

Dunque nel primo capitolo si spiega come ottenere l'IDE, mentre nel secondo come ottenere sull'IDE per la prima volta i progetti JAVA già sviluppati da altri utenti e sui quali si dovrà lavorare e il terzo capitolo tratterà alcuni comandi di GIT.

INTRODUZIONE AD ADOpT

Di seguito é riportata una brevissima introduzione al software che si sta sviluppando in JAVA: ADOpT. ADOpT é un software in fase di sviluppo interamente scritto in JAVA con lo scopo di essere un supporto per le fasi concettuali e preliminari del progetto di un aeromobile, sviluppato nel Dipartimento di Ingegneria Industriale della Federico II. ADOpT si configura come l'evoluzione di ADAS che era scritto in Visual Basic. La scelta dell'utilizzo di JAVA come linguaggio di programmazione é dovuta al fatto che é un linguaggio molto diffuso e si presta bene alla modularitá.

Lo scopo é quello di creare un software completo per supportare la progettazione di un aeromobile e quindi di procedere implementando ed integrando i vari moduli necessari. ADOpT puó essere lanciato o tramite riga di comando oppure tramite interfaccia grafica (GUI). Per effettuare piú prove sui dati e verificare i risultati, in fase di progettazione si utilizza la non gui mode.

Capitolo 1

Utilizzo dell'IDE

L'IDE é un ambiente di sviluppo nel quale scriveremo il nostro codice JAVA. Esistono vari IDE, quello che utilizzeremo sarà Eclipse.

Nello sviluppo condiviso di un software é bene che tutti gli sviluppatori abbiano la stessa versione del software.

Per scaricare Eclipse si vada sul sito www.eclipse.org si clicchi su download scegliendo la versione adatta al proprio PC. Si otterrà così un file zippato che dovrà essere decompresso per poi procedere con l'installazione cliccando semplicemente sull'eseguibile *eclipse.exe* all'interno della cartella.

Al fine di poter eseguire il compilatore Java e la Java Virtual Machine “manualmente” tramite il prompt dei comandi è necessario modificare la variabile di sistema PATH eseguendo le seguenti operazioni:

1. Identificare la directory in cui è stato installato il JDK. In particolare, è necessario conoscere il nome della directory che contiene il file “javac.exe”. Ricercarlo e copiare il percorso.
2. All'interno del menu start fare click con il tasto destro su “Computer” e selezionare “Proprietà” (in alternativa, “Pannello di Controllo” — > “Sistema e sicurezza” — > “sistema”).
3. Selezionare “Impostazioni di sistema avanzate”
4. Selezionare “Variabili d'ambiente”
5. Tra le “Variabili di sistema” (la seconda lista) individuare “Path” e selezionare la stessa tramite un doppio click
6. Il campo “Valore variabile” contiene una lunga lista di directory separate da ; Aggiungere in fondo a tale lista un nuovo ; e la directory identificata in precedenza.
7. Per verificare che tutto funzioni aprire il prompt dei comandi (“Menù start”->”tutti i programmi”->”Accessori”->”Prompt dei comandi”) e digitare il comando “ java -version”.

il sistema dovrebbe rispondere con tre righe simili a:

“ java version “1.7.0_40”

Java(TM) SE Runtime Environment (build 1.7.0_40 – b43)

Java HotSpot(TM) Client VM (build 24.0 – b56, mixed mode, sharing)”

All'avvio Eclipse chiederá quale deve essere la cartella ove salvare i file. Si puó lasciare quella di default oppure definirne una nuova. La prima schermata che uscirá sará quella relativa al tutorial che puó essere chiusa con la X.

La schermata di Eclipse si presenterá divisa in piú finestre la cui organizzazione é chiamata Perspective. Una Perspective rappresenta una configurazione delle finestre dell'IDE. Ciò é dovuto al fatto che Eclipse offre molte funzionalità e le varie perspective già predefinite sono organizzate per poter offrire tutto ciò che serve in base alle esigenze. Ad esempio la perspective Java sará quella che possiamo usare per lo sviluppo del software. Per cambiare o personalizzare le perspective Windows -- > Perspective.

Per una guida sulle basi dell'utilizzo di Eclipse http://www.youtube.com/watch?v=B1YU88UE_7A&list=PL0qAPtx8YtJe2dpE7di4aPJwrQuRD6IDD&index=6

Capitolo 2

Gestione di file condivisi

2.1 Introduzione

In questo capitolo sarà presentato passo per passo come ottenere i file di altri utenti che lavorano al progetto e che sono stati caricati in rete. Per fare questo é necessario preliminarmente importare i file dal web e successivamente caricarli nel nostro IDE.

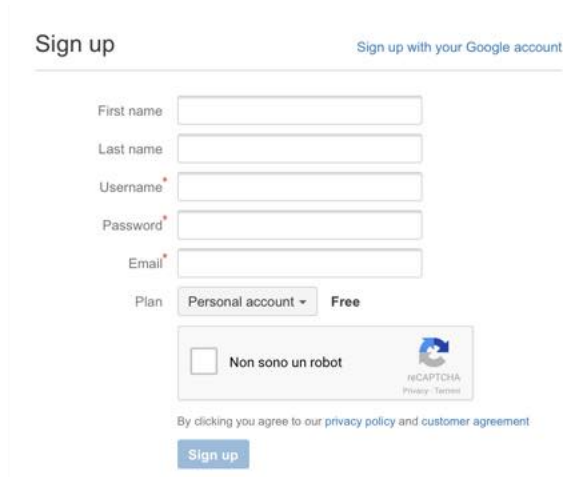
2.2 Prerequisiti

Per poter effettuare i passaggi riportati nei paragrafi che seguono é necessario che siano stati svolti i seguenti step

- Essere in possesso di un IDE
- Creare un account personale su Bitbucket
- Scaricare GIT
- Il primo passaggio é spiegato nel primo capitolo.
- Per quanto riguarda la creazione di un account é necessario andare sul sito <https://bitbucket.org> e registrarsi gratuitamente seguendo le istruzioni.



Figura 2.1: Cliccare su get started for free.



The image shows a 'Sign up' form for a repository. At the top right, there is a link 'Sign up with your Google account'. The form contains the following fields: 'First name', 'Last name', 'Username' (marked with a red asterisk), 'Password' (marked with a red asterisk), and 'Email' (marked with a red asterisk). Below these fields is a 'Plan' section with a dropdown menu set to 'Personal account' and a 'Free' button. Underneath is a reCAPTCHA widget with the text 'Non sono un robot' and a checkbox. At the bottom, there is a line of text: 'By clicking you agree to our [privacy policy](#) and [customer agreement](#)'. A blue 'Sign up' button is located at the bottom center.

Figura 2.2: Compilare tutti i campi necessari.

In seguito si riceverà una mail con l'indirizzo per confermare l'iscrizione. A questo punto si è semplicemente in possesso di un Account.

Per poter accedere agli archivi a cui si è interessati, occorre ricevere l'invito. Tramite l'invito è possibile o solo leggere o anche scrivere (in base ai permessi concessi) sui file in repository.

Dopo che è stato ricevuto l'invito via mail, è possibile accedere alle repository.

- A questo punto occorre ottenere il software di gestione di comandi GIT. Per fare il download bisogna andare sul sito ufficiale <https://git-scm.com/> e cliccare su download e scegliere la versione a 32 o 64 bit.

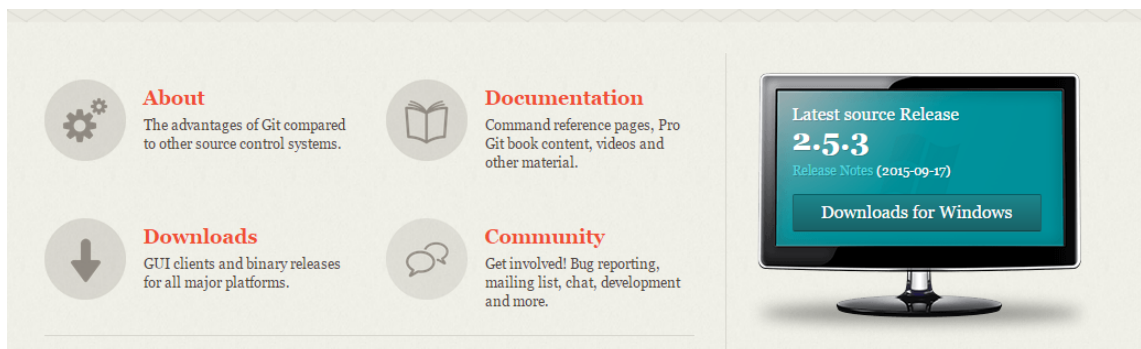


Figura 2.3: Cliccare su Downloads o Download for Windows.

A questo punto per installarlo seguire le impostazioni di Default. In seguito occorrerà lavorare tramite la Git Bash

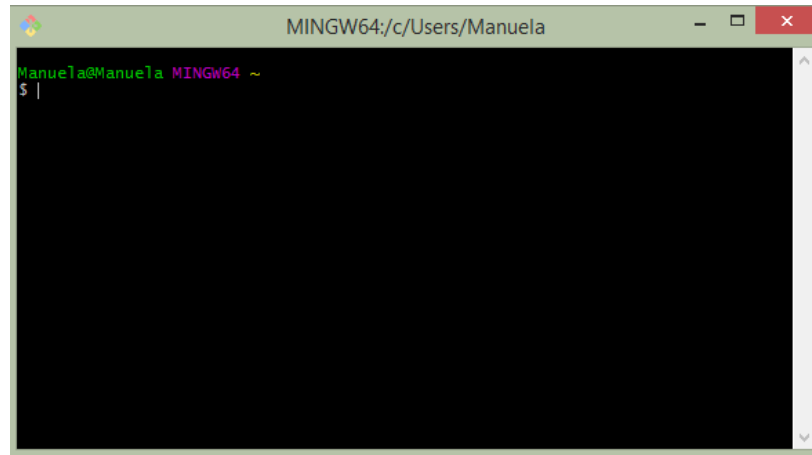


Figura 2.4: Esempio di GIT Bash.

2.3 Download dei file in repository

1. Prima di tutto si consiglia di creare una directory di lavoro che non dovrà essere spostata in seguito. Questa cartella conterrà le sottocartelle, ognuna delle quali corrisponde ad una delle repository su cui si lavora. Un esempio di directory di lavoro può essere la seguente: C: — — — > Users — — — > Nome — — — > NOMECARTELLAPROGETTO. È preferibile non utilizzare directory con spazi.
2. In seguito si apra la GIT bash e ci si deve posizionare nella cartella di lavoro creata nel punto 1. Per fare ciò il comando da utilizzare è **cd** seguito dalla directory. Ad esempio “`cd /c/users/Nome`”
3. Aprire la repository di bitbucket dal sito online e copiare il link come nella figura che segue. Si noti che di fianco al link c'è scritto o HTTPS oppure SSH. Questo è il protocollo con il quale è stato fatto il trasferimento. In questa guida è riportato il procedimento per il protocollo HTTPS, il protocollo SSH serve per condividere file grandi in sicurezza. Va abilitato ed è protetto da password.
4. Scrivere in GIT — — — > “`git clone`” e Incollare il link copiato in Bitbucket . In questo modo nella cartella creata ci sarà un clone dei progetti.
5. Attendere il download.
6. Ripetere dal punto 3 con tutte le repository che servono.

Per quanto riguarda il progetto ADOpT le repository sono due: JPAD e ADOpT. Occorre avere i permessi per leggere e scrivere su entrambe. Il clone dei file va fatto una sola volta, dopo le modifiche saranno gestite con i comandi push e pull spiegati nel prossimo capitolo.

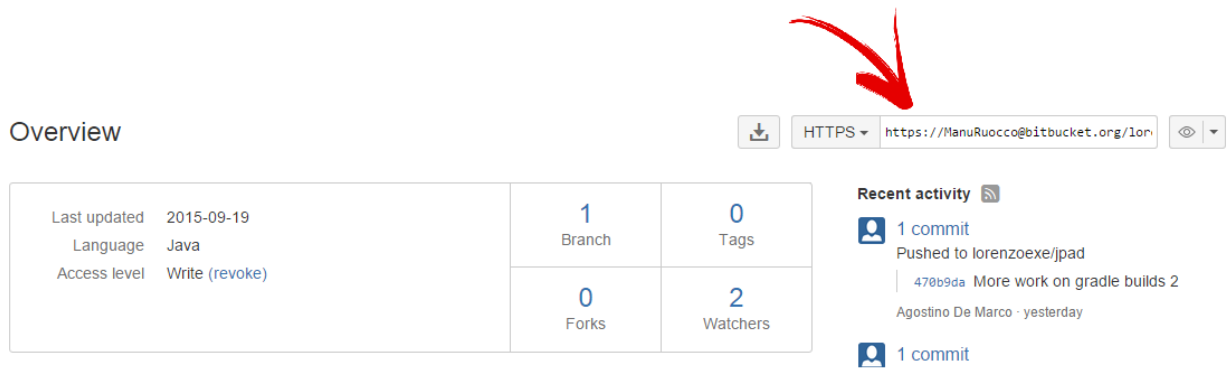


Figura 2.5: Link su Bitbucket per il clone tramite GIT.

2.4 Abilitare le Notifiche

Per abilitare le notifiche di Bitbucket, e quindi ricevere via mail comunicazioni riguardo eventuali modifiche da parte di altri utenti delle repository, è necessario seguirle. In seguito si imposterà la mail per la ricezione delle stesse.

1. Entrare nell'account Bitbucket.org
2. Cliccare su Repository in alto e scegliere quella che si vuole seguire.
3. Entrati nella repository cliccare sulla freccia accanto l'occhio e poi su “ watching this repository”.
4. Cliccare in alto a destra sull'icona dell'Account -- > Manage Account.
5. Nel pannello a sinistra cliccare su Notification.
6. Impostare l'indirizzo email al quale ricevere notifiche.
7. Per gestire di quale repository ricevere notizie cliccare su “ manage” nella sezione “ You are watching”.

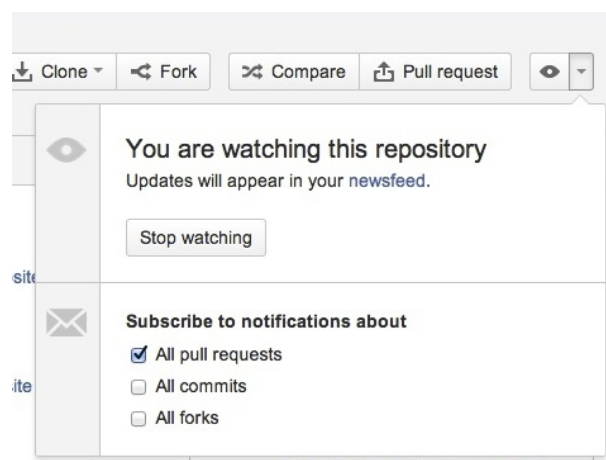


Figura 2.6: Punto 3

2.5 Importare i progetti in Eclipse

Dopo aver fatto il clone dei progetti tramite GIT questi dovranno trovarsi in delle cartelle all'interno della directory definita nel punto 1 del paragrafo precedente. A questo punto occorre importarli in Eclipse:

1. Aprire Eclipse.
2. file --- > import --- > general --- > Existing Project into Workspace --- > selezionare la cartella del file cliccando su Browse --- > in Options cliccare su search for nested projects così che compariranno tutti i file nella cartella e non bisogna importare i file uno per uno.
3. Ripetere dal punto 2 per tutte le repository

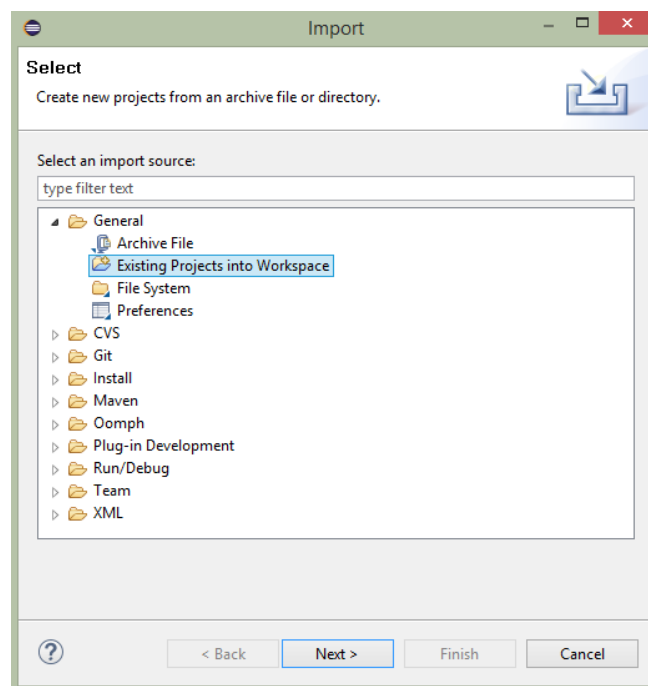


Figura 2.7: Passo 2.

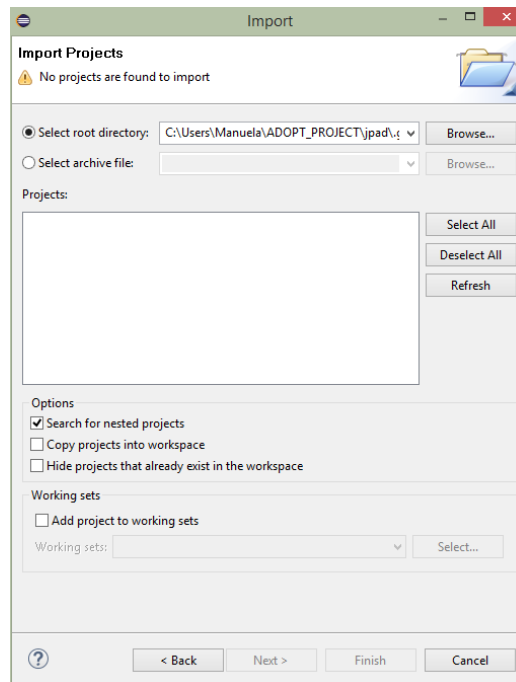


Figura 2.8: Passo 2.

2.6 Abilitare le librerie

A questo punto se si provasse a far partire il file main (ADOPT --- > *it.unina.adopt.main* --- > *ADOPT_GUI.java*) il compilatore darebbe errore. Questo perché occorre ancora fare qualche passaggio per poter utilizzare le librerie che sono richieste nel programma. JAVA fa riferimento ad alcune librerie interne, che sono di default già caricate, e librerie esterne che vanno importate. Per questo motivo occorre abilitare sul proprio IDE tutte le librerie che sono state utilizzate nel codice.

1. Windows --- > preferences --- > java --- > build path --- > classpath variables
2. A questo punto occorre aggiungere tre nuove librerie per ognuna delle quali occorre scrivere il nome e l'indirizzo. L'indirizzo va cercato con folder
 - **Name** *LIBRARY_FOLDER*
Path C : */Users/Manuela/ADOPT_PROJECT/jpad/libraries* (va nella cartella libraries della cartella jpad che è uno dei due file clonati dalle repository tramite GIT)
 - **Name** *ECLIPSE_FOLDER*
Path C : */ProgramFiles/eclipse/plugins* (va nella cartella plugins della cartella di eclipse)
 - **Name** *JAVA_JDK*
Path C : */ProgramFiles/Java/jdk1.8.0_60* (va nella cartella relativa al jdk plugins della cartella di java)

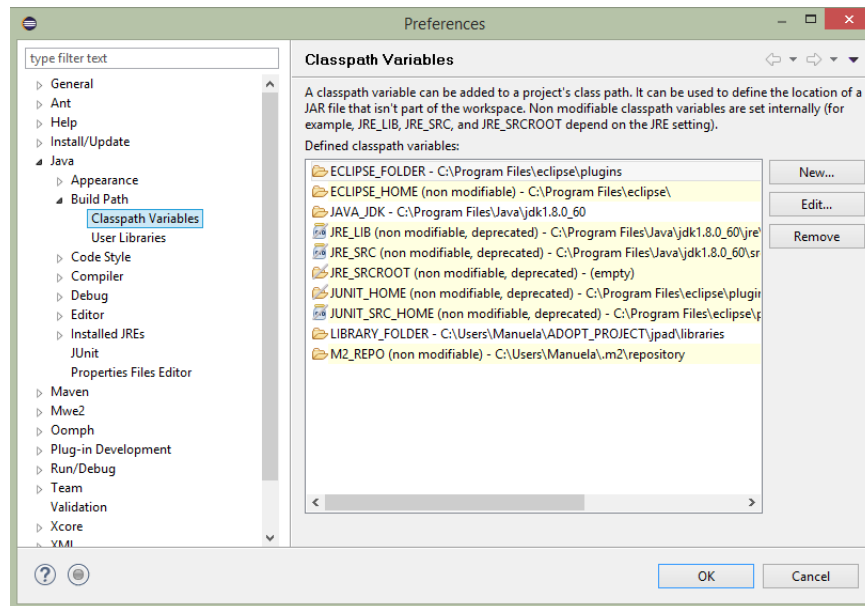


Figura 2.9: Passo 1.

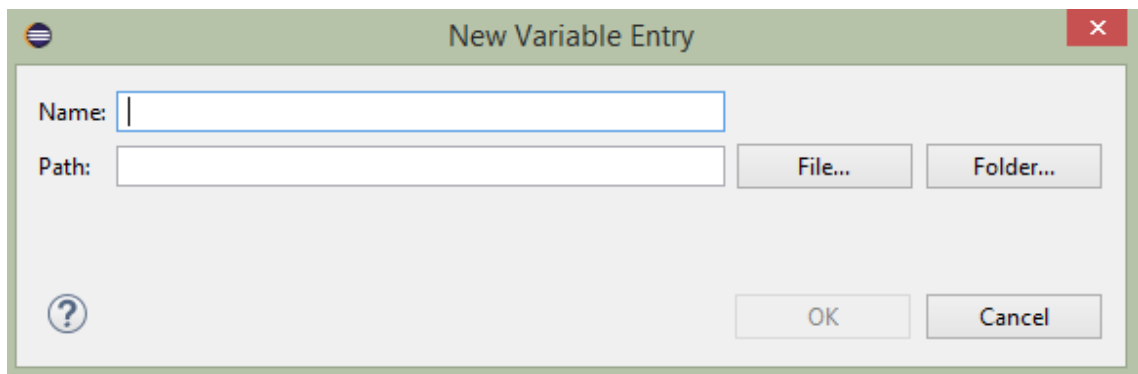


Figura 2.10: Nuove librerie.

2.7 Settare la Run Configuration

La Run Configuration é l'insieme di argomenti, variabili ecc. da passare al programma che si vuole eseguire. Occorre definire una Run Configuration per ogni Main che si avvia. La configurazione sar  salvata e quando la si vuole avviare di nuovo si deve semplicemente cliccare su quella gi  creata in precedenza. Per aprire la finestra di configurazione della Run Configuration

Run — — — > Run Configuration — — — > Java application

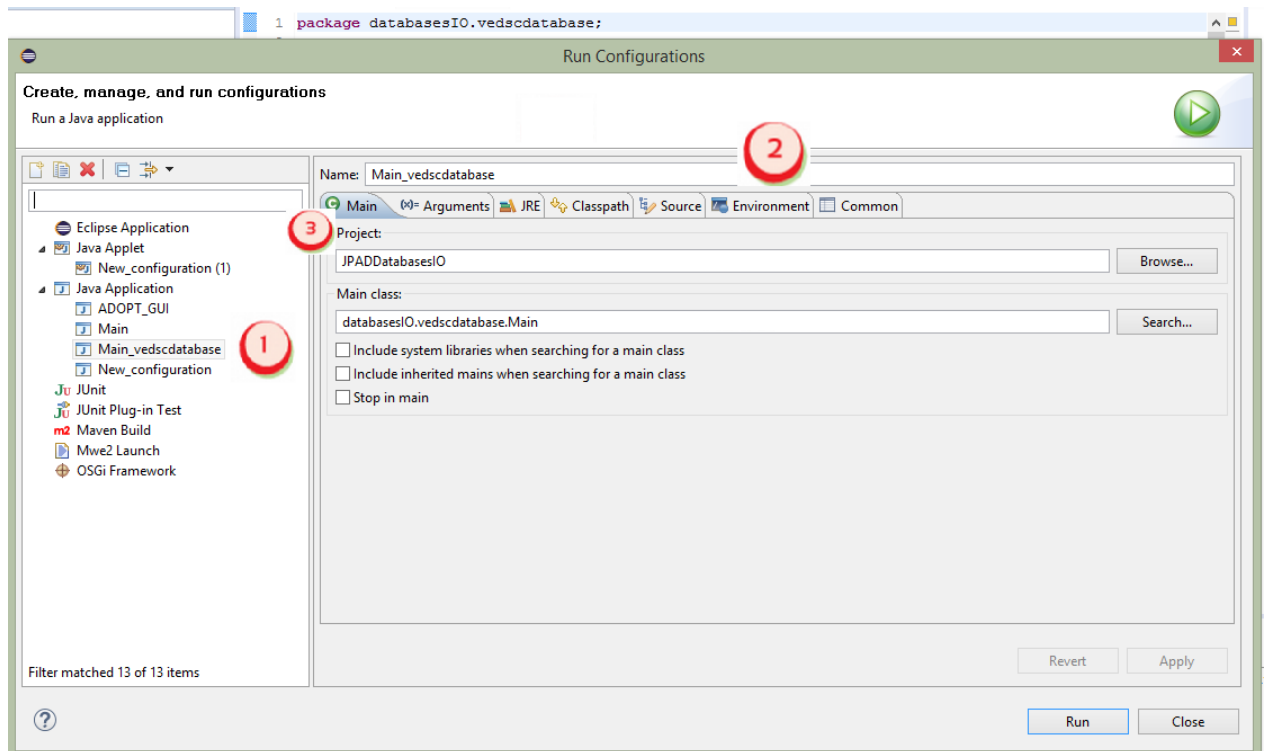


Figura 2.11: Finestra della Run Configuration.

2.7.1 Settare variabili d'ambiente

Per poter utilizzare ADOpT bisogna creare le seguenti variabili d'ambiente:

1. Aprire la finestra della Run Configuration.
2. Cliccare con il tasto destro e premere “ New” .
3. Cliccare sulla configurazione creata.
4. Cliccare su Environment e inserire tre nuove variabili con New...
 - **Variable Path**
Value C : `/Users/Manuela/ADOPTPROJECT/jpad/libraries/hdf`
 - **Variable occjavaDir**
Value C : `/Users/Manuela/ADOPTPROJECT/jpad/libraries/occjava/Win64`
 - **Variable MMGT_OPT**
Value 0

Settare la Run Configuration, come abbiamo fatto in questo caso é importante farlo per ogni Main che si vuole eseguire. Poiché le variabili d'ambiente devono essere sempre definite, si può creare una Run Configuration come é stato appena fatto, per un main e duplicarla per un fare il run di un altro main. In pratica ogni main che andiamo ad eseguire necessita della definizione di una run configuration ove c'è la definizione delle dette variabili d'ambiente, invece di ricreare la configurazione e definire tutte le

variabili si può duplicare quella esistente e cambiare ciò che serve.
Per duplicare una configurazione:

Run --- **>** **Run Configuration** --- **>** **clickare sulla configurazione da duplicare** --- **>** **Duplicate**

Quindi per far eseguire altri Main si duplichi semplicemente una configurazione creata e poi si cambieranno dei valori, lasciando le variabili inalterate. I paragrafi successivi esprimono i porcedimenti per quelle impostazioni che vanno cambiate di Main in Main.

2.7.2 Project e Main Class

1. Cliccare sulla configurazione che si vuole definire e su Main
2. Come Main Class inserire **nomePackage.nomeClasseconMain**
3. Come Project inserire **Main_nomeclasse**. In realtà il nome del progetto può essere qualsiasi ma è bene metterne uno che ricordi il main che si esegue.

2.7.3 Argomenti

Nella sezione “Arguments” della Run Configuration vi è la possibilità di immettere argomenti (in **Program arguments**) che saranno passati direttamente al Main. La stringa che definisce il main in un progetto è

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

Quindi gli argomenti di input nel Main sono un array di un numero qualsiasi di stringhe che sono appunto gli argomenti che possiamo passare al programma tramite la Run Configuration. Questi argomenti possono passare dei valori al progetto che sono definiti nel corpo dello stesso ove è definita anche la sintassi da utilizzare.

2.8 Aggiornamenti di grandi moli di dati

Se da un altro utente sono state fatte grandi modifiche al progetto, come ad esempio una riorganizzazione delle cartelle dei file, queste non saranno lette in automatico da Eclipse. Per poter aggiornare il progetto è possibile fare come scritto di seguito.

1. Rinominare la cartella ove sono salvati i file
(Users --- **>** Nome --- **>** *ADOPT_PROJECT*)
2. Creare una nuova cartella chiamata *ADOPT_PROJECT*
3. Fare nuovamente il pull delle directory jpad e adopt nella cartella *ADOPT_PROJECT*

4. Al termine aprire Eclipse, nel Package Explorer tasto destro, import, existing projet into workspace e selezionare jpad e adopt separatamente
5. F5 per aggiornare

Capitolo 3

Utilizzo di GIT

3.1 Alcuni comandi base

Comando	Significato
\$ cd indirizzo	Per andare in una directory
\$ cd -	Va all'indirizzo precedente
\$ cd ..	Per salire di un livello
\$ ls	Per andare in una directory
\$ mkdi FolderName	Crea una cartella nella directory corrente
\$ mkdir percorso/FolderName	Crea una cartella nella directory specificata
\$ rm - FileName	Per cancellare un file

3.2 Push e Pull

Prima di ogni sessione di lavoro é bene aggiornare lo stato dei file che si hanno sul pc perché possono essere state fatte eventuali modifiche da altri utenti che accedono alla repository.

Per prelevare dalla repository l'ultima versione del file:

1. Aprire la bash di GIT.
2. Andare nella cartella della quale si vuole prendere l'ultima versione (oJPAD o ADOPT)
3. Scrivere il comando “ **\$ git pull**”
4. Quando si apre Eclipse a seguito di un Pull, per aggiornare i file — > F5.

A seguito delle modifiche per caricare in repository l'ultima versione

1. Salvare le modifiche in Eclipse.

2. Aprire la bash di GIT.
3. Andare nella cartella del progetto.
4. Si consiglia di fare sempre un pull prima del push, quindi occorre scrivere il comando “ **\$ git pull**”
5. Scrivere il comando “ **\$ git add**” per aggiornare i file con le modifiche fatte.
Per aggiornare tutti i file della cartella si faccia “ **git add -all**” altrimenti al posto di -all si inserisca il nome del file
6. Se si vuole scrivere un commento che chiarisce ciò che si é modificato (é bene farlo), si può scrivere “**git commit -am “Messaggio”**”
7. Scrivere il comando “ **\$ git push**”
8. Inserire la password di Bitbucket

La prima volta che si fa un push occorre autenticarsi. Occorre scrivere in git le seguenti righe di comando ove al posto di NomeUtente e Email mettere il proprio user e la mail di Bitbucket:

- `git config --global user.name NomeUtente`
- `git config --global user.email Email`

Capitolo 4

Organizzazione del Lavoro: svolgimento dei test.

4.1 Introduzione

Con l'introduzione di un nuovo programmatore nel progetto é necessario creare un nuovo “ *package*” nella sandbox. É consuetudine nominarlo con le iniziali di nome e cognome del programmatore.

Le prime classi programmate saranno interamente fatte nella propria sandbox per evitare di arrecare danni al codice preesistente, in seguito, se verranno implementate nuove classi in JPAD oppure verranno modificate quelle esistenti, all'interno della propria sandbox saranno svolti tutti i test relativi.

Per creare una nuova sandbox:

1. Aprire in Eclipse, Package Explorer -- > JPADSandBox -- > src -- > sandbox
2. Tasto destro su “ sandbox” -- > new -- > package
3. Per aggiungere nuove classi, tasto destro sul proprio package -- > new -- > class

Si consiglia di nominare le classi di test come Test_inizialipackage_numerotest_argomento.

4.2 Come effettuare i test

Per poter effettuare i test é necessario operare su un certo velivolo e utilizzare i relativi dati come input del test. Per ottenere il velivolo é possibile procedere principalmente in due modi:

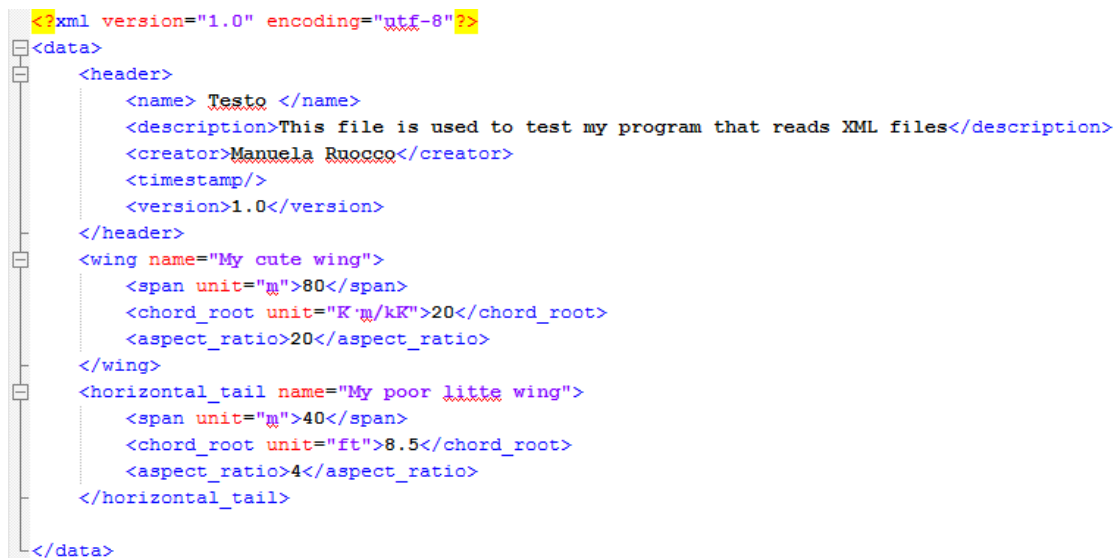
- Leggere da un file di input .xml
- Generare un default Aircraft

4.3 Lettura di dati da un xml

L'Extensible Markup Language (XML) è un metalinguaggio, ovvero un insieme di regole base utilizzate per creare altri linguaggi che consta fondamentalmente nell'utilizzo di tag (marcatori) all'interno dei quali inserire in maniera strutturata il contenuto informativo da rappresentare.

Per prima cosa vedremo le regole fondamentali per scrivere un file di testo .xml che sarà il tipo di file che darò in input al programma che stiamo sviluppando in JAVA. I concetti fondamentali che vedremo sono quelli di :

- Etichetta
- Attributi
- Struttura ad albero
- Intestazione



```
<?xml version="1.0" encoding="utf-8"?>
<data>
  <header>
    <name>Testo </name>
    <description>This file is used to test my program that reads XML files</description>
    <creator>Manuela Ruocco</creator>
    <timestamp/>
    <version>1.0</version>
  </header>
  <wing name="My cute wing">
    <span unit="m">80</span>
    <chord_root unit="K m/kK">20</chord_root>
    <aspect_ratio>20</aspect_ratio>
  </wing>
  <horizontal_tail name="My poor litte wing">
    <span unit="m">40</span>
    <chord_root unit="ft">8.5</chord_root>
    <aspect_ratio>4</aspect_ratio>
  </horizontal_tail>
</data>
```

Figura 4.1: Esempio di un file .XML.

Etichette

Ogni parte di testo è contenuta tra un'etichetta di apertura e una di chiusura che conferiscono un significato al testo. L'etichetta di apertura è una parola chiave, cioè un tag tra due parentesi angolari, mentre l'etichetta di chiusura contiene un carattere di slash all'interno prima della parola chiave.

```
<name>Test XML</name>
```

Le parole chiave che costituiscono le etichette non sono codificate nel senso che è possibile mettere una qualsiasi parola. Lo scopo è mettere una parola che chiarisca il significato del testo contenuto all'interno. I nomi dei tag non possono contenere spazi o iniziare con i numeri ma possono contenere il carattere di *underscore*.

Attributi

Le etichette possono anche avere degli **attributi**. Gli attributi sono uno strumento per introdurre informazioni aggiuntive all'interno di un elemento. Essi non influiscono sul contenuto dell'elemento stesso, poiché si trovano all'interno dei tag di apertura o in un tag di elemento vuoto. Il valore che assume l'attributo è racchiuso tra virgolette e segue il nome dell'attributo.

```
<tag attribute1='value' attribute2='value'> text </tag>
```

Struttura ad Albero

Un file xml deve avere una struttura “ad Albero” nel senso che ci sono nodi esterni che si diramano in nodi interni creando dei percorsi.

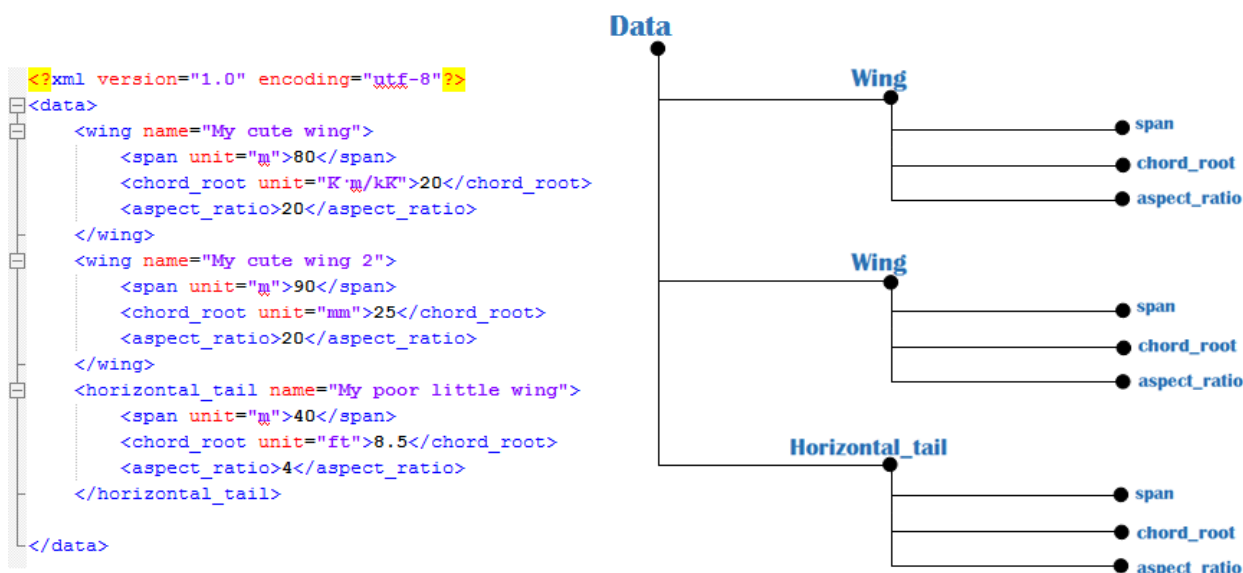


Figura 4.2: Struttura ad albero di un file XML.

Quindi il percorso Data —> wing —> span restituisce i valori 80 e 90, essendoci due percorsi con lo stesso indirizzo.

Intestazione

La prima riga indica la versione di XML in uso e specifica la codifica UTF-8 per la corretta interpretazione dei dati.

```
<?xml version="1.0" encoding="utf-8"?>
```

4.3.1 Caratteri speciali in xml

La seguente tabella riassume le sequenze da utilizzare per definire dei caratteri speciali come testo in un file XML.

Entità	Carattere corrispondente
&	&
<	<
>	>
"	"
'	'

Figura 4.3: Caratteri speciali in XML.

4.4 Input da File

In questo paragrafo vedremo come leggere file XML e rendere utilizzabili i dati. A seguito della lettura di un file, infatti, può essere necessario memorizzare i dati in apposite variabili per poterli utilizzare all'interno del programma.

4.4.1 Scrivere e salvare il file

La prima cosa da fare, chiaramente, è quella di creare il file. Il file può essere creato seguendo gli step del passaggio precedente modificando a proprio piacimento i dati che si vogliono mettere. Si noti che l'utilizzo di attributi è molto comodo per definire le unità di misura della grandezza.

Dopo che è stato creato il file questo andrà salvato in una cartella di lavoro.

Per convenzione i file di input di ADOPt sono salvati nella cartella “in” all'interno del progetto che si sta utilizzando.

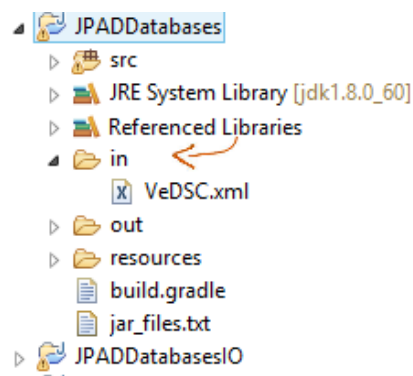


Figura 4.4: Cartella ove salvare il file .XML. In questo caso il progetto utilizzato è JPADDATABASE

Per richiamare il file all'interno del programma, occorre capire che la ricerca del file parte dalla cartella del progetto. Pertanto se il file di input viene messo nella cartella del progetto (come JPADDATABASE) senza nessuna sottocartella (scelta sconsigliata) allora per richiamarlo basta dare il nome del file e l'estensione. Se, invece, viene messo

in una sottocartella (come la cartella “in”) del progetto , allora occorre dare il nome della cartella e del file. Il seguente schema riassume quanto detto.

- File nella cartella del progetto — — — > *nome.xml*
- File in una sottocartella della cartella del progetto — — — > *sottocartella\nome.xml*

4.4.2 Leggere un file xml

Per leggere un file xml che é stato passato come argomento del main si possono utilizzare i seguenti comandi. Prima occorre leggere il file e per ottenere i valori in un certo percorso si da quest’ultimo come stringa.

```
//Start Reading XML file

Test_MR_03_Downwash theObj = new Test_MR_03_Downwash();

theObj.theCmdLineParser.printUsage(System.out);
theObj.manageCmdLineArguments(args);

String xmlFilePath = theObj.get_inputFile().getAbsolutePath();

System.out.println("\n INPUT FILE, full path: " + xmlFilePath + "\n");

JPADXmlReader xmlReaderUnits = new JPADXmlReader(xmlFilePath);
```

Queste stringhe di codice servono per consentire la possibilità di leggere, mentre per ogni valore che si vuole leggere occorre fare come segue:

```
//Sweep Angle at c/4
Amount<?> sweepQuarterChordEqAmountDeg = xmlReaderUnits.getXMLAmountWithUnitByPath("//Equivalent_Wing_parameters/sweepc4").to(ST.RADIAN);
```

variabile in cui andrà il valore letto

classe che gestisce la lettura

metodo che consente la lettura di un amount

percorso

4.5 Generare un default Aircraft

Il codice JPAD consente di generare semplicemente un aereo di default che può essere l’ATR-72 come turboProp, oppure il B747-100B come TurboFan. I dati di questi velivoli sono scolpiti nel codice ed altri sono calcolati a partire dagli stessi. Per definire un aereo di default su cui fare le analisi occorre seguire la seguente routine:

```

System.out.println("Downwash gradient test calculator");
System.out.println("-----");

// Assign all default folders
MyConfiguration.initWorkingDirectoryTree();

// Setup database
String databaseFolderPath = MyConfiguration.getDir(FoldersEnum.DATABASE_DIR);
String aerodynamicDatabaseFileName = "Aerodynamic_Database_Ultimate.h5";
AerodynamicDatabaseReader aeroDatabaseReader = new AerodynamicDatabaseReader(databaseFolderPath, aerodynamicDatabaseFileName);

// Operating Condition / Aircraft / AnalysisManager (geometry calculations)
OperatingConditions theCondition = new OperatingConditions();
Aircraft aircraft = Aircraft.createDefaultAircraft();
aircraft.get_theAerodynamics().set_aerodynamicDatabaseReader(aeroDatabaseReader);
aircraft.set_name("ATR-72");

ACAnalysisManager theAnalysis = new ACAnalysisManager(theCondition);
theAnalysis.updateGeometry(aircraft);

```

Figura 4.5: Generazione di un default aircraft in JPAD.

