CORSO DI LAUREA TRIENNALE IN INFORMATICA (GRUPPO 2)

CORSO DI OBJECT ORIENTATION

Docente: Prof. Porfirio Tramontana

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione (DIETI)

Via Claudio 21, stanza 4.06

Università degli Studi di Napoli Federico II

e-mail: porfirio.tramontana@unina.it

Website: https://sites.google.com/view/porfiriotramontana/home-page

tel. 081 768 3901

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
- Il Linguaggio Java
- Realizzazione di interfacce grafiche ad eventi.
- Metodologie di Sviluppo

I PARADIGMI DI PROGRAMMAZIONE

- Forniscono la filosofia e la metodologia con cui si scrivono i programmi
- I linguaggi devono *consentire* ma soprattutto *spingere* all'adozione di un particolare paradigma
  - Procedurale
  - Funzionale
  - Dichiarativo
  - Orientato agli oggetti

PARADIGMA IMPERATIVO/PROCEDURALE

- Enfasi sulla soluzione algoritmica dei problemi mediante modifica progressiva dei dati in memoria
  - Esecuzione sequenziale di istruzioni
  - Cambiamento dello stato di memoria (le variabili) tramite assegnamento
- Aderenti al modello della macchina di von Neumann
- Molto efficienti
- Ha mostrato limiti nello sviluppo e manutenzione di sw complessi
- I linguaggi imperativi: Pascal, C

PARADIGMA DICHIARATIVO

- Specificare le caratteristiche della soluzione con dichiarazioni che le descrivono piuttosto che il procedimento per ottenerla
  - Il programma viene eseguito da un risolutore di problemi che determina la strategia migliore
  - Paradigma utilizzabile in campi specifici
- Esempi:SQL
  - Recupera dal database tutti gli studenti con matricola che inizia per «N86»

PARADIGMA FUNZIONALE

- La computazione avviene tramite funzioni che, applicate ai dati, riportano nuovi valori
  - Ogni funzione è un modulo a sé dipendente unicamente dal valore dei suoi argomenti
  - L'effetto globale è ottenuto concatenando opportunamente funzioni anche richiamando sé stesse (ricorsione)
  - Modello che si rifà alla teoria delle funzioni ricorsive
  - Scarso supporto ai costrutti di ripetizione tramite iterazione
  - Di recente utilizzato anche nel contesto di sistemi ad eventi (ad esempio interfacce grafiche)
- Lisp, ML, Scala

PARADIGMA A OGGETTI

- Evoluzione dell'imperativo/procedurale, per facilitare la programmazione.
- Pone al centro delle attività la *modellazione* del dominio del problema e successivamente la sua soluzione, definendo gli *oggetti* coinvolti e raggruppandoli in *classi*
- Ogni classe avrà delle proprie *responsabilità* alle quali corrispondono funzioni che andranno risolte con *metodi* che implementaranno *algoritmi*
- Gli *algoritmi non sono più il centro del processo di sviluppo ma solo una sua parte*
- Obiettivo: migliorare l'efficienza del processo di produzione e mantenimento del software, rendendolo adatto allo sviluppo di applicazioni di medie e grandi dimensioni

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
- Il Linguaggio Java
- Realizzazione di interfacce grafiche ad eventi.
- Metodologie di Sviluppo

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
  - Concetti di astrazione dei dati, di definizione di tipi personalizzati, e di incapsulamento
  - Oggetti, variabili, classi e metodi
  - Costruttori
  - Ereditarietà e polimorfismo
  - Overloading e Overriding
  - Le interface e loro benefici.
  - This e Super
- Il Linguaggio Java
- Realizzazione di interfacce grafiche ad eventi.
- Metodologie di Sviluppo

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
- Il Linguaggio Java
  - JVM.
  - JDK
  - Tipi Primitivi e Riferimenti in Java.
  - Garbage Collector
  - Le collezioni in Java.
  - Le classi Wrapper in Java. Autoboxing e Unboxing
  - Le Eccezioni in Java.
- Realizzazione di interfacce grafiche ad eventi.
- Metodologie di Sviluppo

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
- Il Linguaggio Java
- Realizzazione di interfacce grafiche ad eventi con AWT e Swing.
  - Programmazione event-driven
  - Differenti tipi di Finestre, pannelli e Component.
  - Architetture per applicazioni con GUI: pattern Boundary – Control – Entity (BCE)
  - Il concetto di classe Controller.
- Metodologie di Sviluppo

CONTENUTI DEL CORSO

- La programmazione orientata agli oggetti
- Il Linguaggio Java
- Realizzazione di interfacce grafiche ad eventi.
- Metodologie di Sviluppo
  - UML: Class Diagrams e Sequence Diagrams.
    - Corrispondenze con codice sorgente.
  - Gli ambienti di Sviluppo Integrati (IDE): Eclipse
  - I Sistemi di Version Control.
    - GIT
  - Le CRC Cards
  - Pattern DAO

OBIETTIVI DEL CORSO DI OBJECT ORIENTATION

- Conoscenze che si intendono trasmettere (sapere):
  - Acquisizione delle competenze di base per la progettazione object-oriented attraverso la comprensione dei concetti di astrazione sui dati, di incapsulamento dell'informazione, di coesione e accoppiamento, e di riutilizzo del codice
  - Comprensione delle differenze tra paradigma object-oriented e il paradigma procedurale
  - Conoscenza del linguaggio Java per una buona progettazione object-oriented e per la promozione del riutilizzo del software

CAPACITÀ APPRESE A VALLE DEL CORSO

- Capacità che si intendono sviluppare (saper fare):
  - Definizione di una strategia risolutiva con un approccio orientato agli oggetti, con la sua implementazione nel linguaggio Java, garantendo il giusto equilibrio tra qualità ed efficienza del software
  - Sviluppo di progetti con attività individuali e di gruppo

ORGANIZZAZIONE DEL CORSO

- ~24 lezioni:
  - Generalmente in presenza, trasmesse e videoregistrate anche su Teams
    - Le modalità di accesso/prenotazione delle aule sono quelli stabiliti per tutto l'Ateneo
  - Esercitazioni su carta o con i portatili, eventualmente a distanza

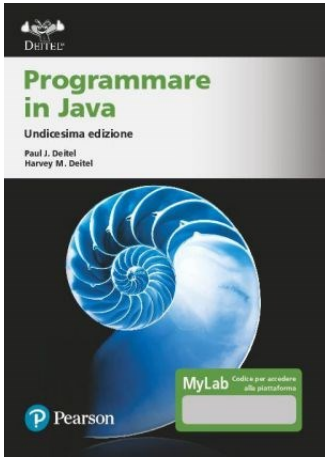
- Orario
  - Mar 08:30 – 10:30 - (posticipato verso 08:45 – 10:45)
  - Gio 16:30 – 18:30 - (anticipato a 16:05 – 18:00)

- Ricevimento studenti:
  - Durante il I semestre, Lunedì 09.30-11.30 nello studio del docente o in caso di necessità su Teams
  - Dopo la fine del semestre, vedere le indicazioni sul sito del docente
    - Nei mesi invernali 2022, martedì mattina, ore 9:30

MATERIALE DIDATTICO

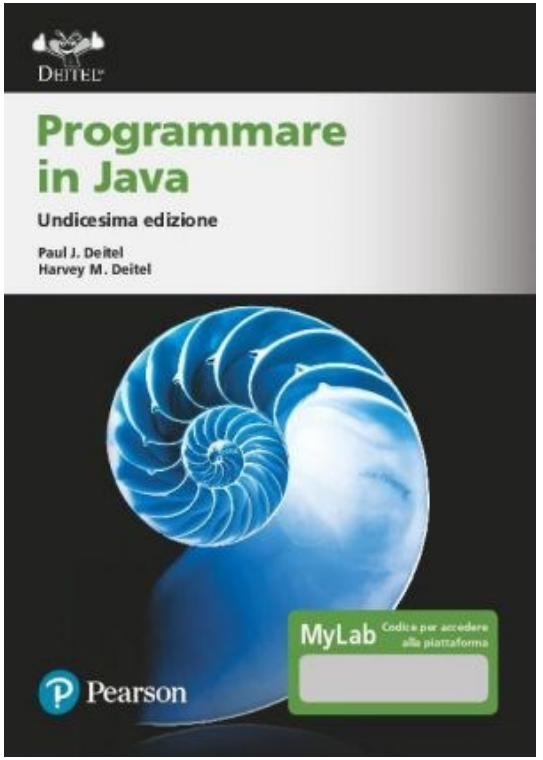
Libri di testo :

- Paul J. Deitel - Harvey M. Deitel: Programmare in Java, 11/Ed., Pearson
- Martin Fowler: UML Distilled, 4/Ed., Pearson



TESTO DI RIFERIMENTO PER JAVA

- Paul J. Deitel - Harvey M. Deitel: Programmare in Java, 11/Ed., Pearson
  - Capitolo 1, sezioni 1.8, 1.9, 1.10
  - Capitolo 2, sezioni da 2.1 a 2.9
  - Capitolo 3 (tranne 3.6)
  - Capitolo 4 (tranne 4.15),
  - Capitolo 5 (tranne 5.11)
  - Capitolo 7, tranne 7.13, 7.17
  - Capitolo 8, sezione 8.4, 8.6, 8.7, 8.8, 8.10, 8.11, 8.13, 8.14
  - Capitolo 9
  - Capitolo 10 da 10.1 a 10.9 e 10.13
  - Capitolo 11, da 11.1 a 11.7 e 11.9
  - Capitolo 14, sezioni 14.1, 14.2, 14.3
  - Capitolo 15, sezioni 15.1, 15.2, 15.4
  - Capitolo 16 da 16.1 a 16.6
- Per chi vuole approfondire la GUI con JavaFX, capitoli 12 e 13



MATERIALE DIDATTICO

Ulteriori libri consigliati:

- Parti generali
  - I. Sommerville. Software Engineering, VIII ed., Addison Wesley, 2007.
- Progettazione ad oggetti
  - C. Larman, Applicare UML e i Pattern - Analisi e Progettazione orientata agli Oggetti, III ed. Prentice-Hall, 2005.
  - B. Bruegge, A. Dutoit. Object-Oriented Software Engineering, Pearson, 2008. (Alternativo a C. Larman).
- UML
  - Stevens Rod Pooley, Usare UML, Addison Wesley, 2008.
  - J. Arlow, Ila Neustadt, UML2 e Unified Process, McGraw-Hill, 2006.
- Altri testi su aspetti specifici
  - P. Amman, J. Offutt. Introduction to software testing, Cambrigde Univerisity Press, 2008.
  - E. Gamma, R. Helm, R.Johnson, J.Vissides. Design patterns, Addison Wesley

MATERIALE DIDATTICO

- Strumenti per le esercitazioni:
  - StarUML (o altri equivalenti) per la modellazione UML
    - http://staruml.sourceforge.net/en/download.php
- Ambiente Eclipse
  - A supporto sia dello sviluppo che della modellazione
    - http://www.eclipse.org/downloads/

COMUNICAZIONI DOCENTE → STUDENTI

- Sito Istituzionale del docente:
  www.docenti.unina.it/porfirio.tramontana

- E' richiesta la registrazione al corso su questo sito
  - Consente di ricevere direttamente mail relative agli avvisi più urgenti

- Lo spazio Teams verrà utilizzato per:
  - Lezioni registrate
  - Materiale didattico (generalmente messo in copia anche su docenti)
  - Appunti e annotazioni sul corso

COMUNICAZIONI STUDENTI → DOCENTE

- Mail :
  - Solo per quesiti brevi!
    - porfirio.tramontana@unina.it
  - Subject: [OO] *e poi l'oggetto*
  - *Utilizzare sempre la mail istituzionale @studenti.unina.it*
  - Firmare SEMPRE le mail
  - Non mandare mail per quesiti su aspetti già descritti nel sito istituzionale e/o nel materiale didattico.
    - Consultare sempre prima il materiale messo a disposizione
- Chat di Teams
  - Solo per comunicazioni urgenti con risposte immediate
- Per quesiti articolati esiste il ricevimento
  - Non è ovviamente ammessa la richiesta di una revisione dell'elaborato d'esame, essendo oggetto della valutazione
- La comunicazione cliente/committente è uno degli aspetti chiave di un professionista dell'informatica!

MODALITÀ DI ESAME

- Progetto obbligatorio di gruppo
  - Progettazione e Implementazione di un piccolo sistema software
  - Presentazione di gruppo al docente del progetto
    - Documentazione + Demo max 10 min.
    - Discussione individuale dell'elaborato
      - (domande poste dal docente ai singoli studenti)
- Scritto: esercizi e domande aperte sull'intero programma.
  - Obbligatorio.

- Voto: Media di Progetto e Scritto

PROGETTO

- 4 – 5 tracce, in comune con il corso di Basi di Dati.
- Assegnazione random a gruppo
- Consistenza numerica: 2 o 3 persone (consigliate 2)
  - Allo scopo di iniziare a maturare esperienza di lavoro in team.
- Formazioni dei gruppi
  - Autonome comunicate nel gruppo del corso
    - Fornirò un modulo apposito
  - Operate dal docente in base alle disponibilità per studenti che non riescano a stabilire formazioni autonome.
- Variazione dei gruppi
  - Ogni variazione di un gruppo ufficializzato deve essere concordata con il docente
- Consistenza con basi di dati
  - Nel caso il gruppo si separi dopo l'esame di basi di dati, gli studenti procedono autonomamente

PROGETTO E PROVA SCRITTA

- I legami di Gruppo terminano con la presentazione dell'elaborato
  - La prova scritta è individuale e non deve essere necessariamente sostenuta contemporaneamente da tutti i membri del Gruppo
- La demo dell'elaborato con la discussione e le domande del docente deve essere sostenuta da tutto il Gruppo contemporaneamente
  - Nel caso in cui un componente non fosse pronto, deve comunicarmi esplicitamente (via mail o chat) la sua rinuncia all'appartenenza al Gruppo
    - Per evitare che un Gruppo si sciolga all'insaputa del componente "ritardatario"
    - Il componente rimasto isolato può riaggregarsi con altri componenti isolati

VALUTAZIONE DEL PROGETTO

- Qualità degli artefatti e della demo.
  - Ogni martedì mattina per tutto il periodo fino all'inizio dei corsi del secondo semestre (inverno 2022)
    - Previa prenotazione via mail con invio del progetto
- **Valutazione dell'intera interazione committente-contraente.**
  - Interazione e uso degli strumenti di comunicazione.
  - Qualità grafica dei documenti prodotti.
  - Qualità della presentazione finale.
- Il progetto può essere consegnato fino a Ottobre 2022.

CHEATING POLICY

- Viene usato uno strumento automatico di Cheating Detection per il confronto di tutto ciò che viene consegnato.

- In caso di due o più progetti siano ritenuti troppo simili, ad insindacabile giudizio dei docenti, il progetto più recente (e entrambi i progetti se consegnati nello stesso appello) sarà/saranno annullato/i e sarà/saranno data una nuova traccia, più estesa e complessa di quella originaria al/ai gruppi che hanno prodotto questo/i elaborato/i