

Programmazione OO (Resto 1)

a.a. 2018/2019

Introduzione al corso

Prof. Carmine Gravino
Email: gravino@unisa.it

Informazioni generali

- **Crediti:** 9 CFU
 - 48 ore Frontali
 - 24 (*2 solo per il docente) ore di Laboratorio/Esercitazioni
- **Orario lezioni:**
 - Lunedì 14-18 (Lab. Reti) – 2 Turni
 - Martedì 15-17 (aula P4)
 - Giovedì 15-17 (aula P4)
- **Sito del corso:**
<http://elearning.informatica.unisa.it/el-platform/>
- **Testo:** Concetti di Informatica e fondamenti di Java, Cay Horstmann. Apogeo

Informazioni generali

- Orario ricevimento:
 - Martedì 9-11
 - Martedì 17-18
 - Giovedì 17-18
 - (contattare il docente via email)

CORSI II ANNO - CLASSE 2 - MATRICOLA CONGRUO A 1

I Semestre - Anno Accademico 2018-2019

ORARIO	LUNEDÌ	MARTEDÌ	MERCOLEDÌ	GIOVEDÌ	VENERDÌ
09 - 10	BD - lab Reti	Ricevimento	SO - Lab P13		SO
10 - 11	BD	Ricevimento	SO - Lab		SO
11 - 12	BD		SO - Lab		BD
12 - 13	BD		SO - Lab		BD
13 - 14		SO - P4		BD	
14 - 15	POO - Lab Reti	SO		BD	
15 - 16	POO - Lab	POO		POO	
16 - 17	POO - Lab	POO		POO	
17 - 18	POO - Lab	Ricevimento		Ricevimento	

Leggenda
SO = Sistemi Operativi (CARPENTIERI)
POO = Programmazione OO (GRAVINO)
BD = Basi di Dati (POLESE)

Informazioni generali (cont.)

- **Durata del corso:**
 - Lunedì 17 Settembre – Venerdì 21 Dicembre 2018
- **Prima prova:**
 - Seconda settimana Novembre 2018
- **Seconda prova:**
 - Ultima settimana del corso

Informazioni generali (cont.)

- **Verifica dell'apprendimento:**
 - gli studenti devono superare un esame finale costituito da una prova pratica in laboratorio ed una prova orale.
 - Una tipica prova pratica richiede la soluzione di esercizi di programmazione in java e la loro implementazione con l'ausilio di un pc, utilizzando l'ambiente integrato di sviluppo software appreso durante il corso.
 - La prova orale consiste in un colloquio sulle esercitazioni effettuate durante il corso, gli esercizi svolti nella prova pratica e gli argomenti in programma.

Syllabus

- **Prerequisiti:** Lo studente dovrebbe avere una buona conoscenza del linguaggio C, delle tecniche di programmazione procedurale e delle problematiche relative alla codifica in codice di algoritmi semplici. Conoscenze di base e discreta familiarità con un sistema operativo (windows, linux,...).

Syllabus (cont.)

- **Conoscenza e capacità di comprensione**

- 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***.

Comprendere le differenze tra questo paradigma e il paradigma procedurale. Sintassi completa del linguaggio di programmazione java. ***Utilizzo del linguaggio java*** per la definizione di classi e per la promozione del riutilizzo del software. Conoscenza delle principali classi messe a disposizione dall'ambiente jdk 7. Conoscenza dei fondamenti teorici del ***polimorfismo*** e del meccanismo ***dell'ereditarietà***. Conoscenza e padronanza dei concetti di classi astratte, ***interfacce*** e del meccanismo di gestione delle eccezioni. Rudimenti di programmazione grafica e ad eventi in java. Realizzazione e consultazione di documentazione di un programma.

Syllabus (cont.)

- **Capacità di applicare conoscenza e comprensione:**
 - capacità di *analisi di problemi*, di specifica dei requisiti e di *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.
- **Autonomia di giudizio:**
 - discussione dei pro e contro delle tecniche e soluzioni studiate. Analisi critica dei lavori progettuali.

Syllabus (cont.)

- **Abilità comunicative:**
 - partecipazione alla formulazione delle strategie risolutive rispetto ai problemi proposti. Presentazione e discussione delle soluzioni individuate.
- **capacità di apprendere:**
 - consultazione individuale della documentazione delle librerie java. Ricerca di classi esistenti e individuazione di nuovi metodi per la soluzione dei problemi proposti.

Contenuti del corso (cont.)

- Introduzione alla programmazione orientata agli oggetti e ai concetti di astrazione dei dati e di incapsulamento.
- Introduzione a java e al processo di compilazione.
- Introduzione al toolkit di programmazione jdk
- Oggetti, variabili, riferimenti; classi e metodi, comunicazione fra oggetti, parametri espliciti e impliciti; il riferimento this.
- Tipi di dati fondamentali.
- Aritmetica e funzioni matematiche.
- Invocazione di metodi statici.
- La classe string; l'input da finestra di dialogo.
- Imparare ad acquisire dati in ingresso da console; conoscere il tipo di dati char; l'input da file.
- File e flussi. Leggere e scrivere file di testo; formato binario e formato di testo

Contenuti del corso (cont.)

- Utilizzo degli array in java. Richiami sui costrutti di controllo.
- Le collezioni in java: l'uso di vettori (array lists) come raccoglitori di oggetti; comprendere quando è il caso di usare vettori e quando array nei nostri programmi; le classi involucro.
- Cenni sui tipi generici, sui tipi enumerativi e sui meccanismi di 'autoboxing' e 'unboxing'.
- Progettazione di classi. Imparare a scegliere le classi appropriate da implementare comprendere i concetti di coesione e accoppiamento; minimizzare l'uso degli effetti collaterali comprendere la differenza tra metodi di istanza e metodi statici; comprendere le regole di portata delle variabili locali e delle variabili istanza; chiamata per valore e chiamata per riferimento; documentare le responsabilità dei metodi e dei loro chiamanti con precondizioni e postcondizioni.

Contenuti del corso (cont.)

- Cenni sull'uso delle schede CRC nella progettazione delle classi. Pacchetti.
- Imparare cosa sono le interfacce; essere in grado di convertire tra riferimenti a classe e a interfacce; comprendere il concetto di polimorfismo; apprezzare come le interfacce possano essere usate per ridurre l'accoppiamento tra le classi; le classi interne.
- Ereditarietà e riuso. Comprendere la relazione di ereditarietà tra classi; la riscrittura di metodi; la ridefinizione di campi; la conversione di tipo tra sottoclasse e superclasse.
- Il polimorfismo e il binding dinamico; le classi astratte; il controllo di accesso; la superclasse universale object.
- Meccanismo di gestione delle eccezioni. Imparare cosa sono le eccezioni controllate e non controllate; il ciclo try-catch-finally; progettare le eccezioni.

Contenuti del corso (cont.)

- Testing: verifica del comportamento degli oggetti; scelta dei casi di test; Integrazione di classi.
- Programmazione grafica e applet.
- Introduzione alle applet e alla programmazione grafica: gestione delle forme, dei font e dei colori; utilizzo di componenti per l'input; primitive per la gestione del contesto grafico; personalizzazione dei pannelli, gestione del layout, gestione dei principali componenti swing.
- Programmare ad eventi: comprendere il modello a eventi di java; installare (o registrare) listener di mouse o di azioni.
- Cenni sul paradigma model-view-controller per lo sviluppo di interfacce grafiche

Workplan del Corso

	Lunedì (ore: 14-18 Lab: Reti)	Martedì (ore: 15-17 Aula: P4)	Giovedì (ore: 15-17 Aula: P4)
Settimana 1 (17-21 settembre)	Introduzione al corso 14-16	Programmazione OO, Java	Programmazione OO, Java
Settimana 2 (24-28 settembre)		Utilizzare gli Oggetti	Realizzare Classi
Settimana 3 (1-5 ottobre)		Tipi di Dati Fondamentali	Decisioni/Iterazioni
Settimana 4 (8-12 ottobre)		Vettori e Array	Progettazione di Classi
Settimana 5 (15-19 ottobre)		Interfacce e Polimorfismo	Interfacce e Polimorfismo
Settimana 6 (22-26 ottobre)		Ereditarietà	Ereditarietà
Settimana 7 (29 ott - 2 novembre)		Testing e Debugging	FESTA
Settimana 8 (5-9 novembre)		Progettazione OO	Gestione delle Eccezioni
Settimana 9 (12-16 novembre)	1^ prova	1^ prova	Gestione delle Eccezioni
Settimana 10 (19-23 novembre)		IO e Flussi	Programmazione Grafica
Settimana 11 (26 -29 novembre)		Gestione degli Eventi	Gestione degli Eventi
Settimana 12 (3-7 dicembre)		Interfacce Grafiche Utente	Interfacce Grafiche Utente
Settimana 13 (10-14 dicembre)	2^ prova	2^ prova	Programmazione Generica - Thread
Settimana 14 (17-21 dicembre)			