



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Matematica e Informatica
ANNO ACCADEMICO OFFERTA	2018/2019
ANNO ACCADEMICO EROGAZIONE	2019/2020
CORSO DILAUREA	INFORMATICA
INSEGNAMENTO	LINGUAGGI DI PROGRAMMAZIONE
TIPO DI ATTIVITA'	B
AMBITO	50166-Discipline Informatiche
CODICE INSEGNAMENTO	04758
SETTORI SCIENTIFICO-DISCIPLINARI	INF/01
DOCENTE RESPONSABILE	FICI GABRIELE Professore Associato Univ. di PALERMO
ALTRI DOCENTI	
CFU	9
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	153
NUMERO DI ORE RISERVATE ALLA DIDATTICA ASSISTITA	72
PROPEDEUTICITA'	05880 - PROGRAMMAZIONE E LABORATORIO C.I.
MUTUAZIONI	
ANNO DI CORSO	2
PERIODO DELLE LEZIONI	1° semestre
MODALITA' DI FREQUENZA	Facoltativa
TIPO DI VALUTAZIONE	Voto in trentesimi
ORARIO DI RICEVIMENTO DEGLI STUDENTI	FICI GABRIELE Martedì 15:00 16:00 Studio 202, DMI. Giovedì 15:00 16:00 Studio 202, DMI.

DOCENTE: Prof. GABRIELE FICI

PREREQUISITI	Conoscenze di programmazione in C, matematica discreta (array, liste, relazioni, calcolo combinatorio), conoscenze di base di logica (calcolo proposizionale, inferenza, tipi di dimostrazione, principio di induzione).
RISULTATI DI APPRENDIMENTO ATTESI	Conoscenza e capacita' di comprensione Conoscenza delle macchine astratte. Conoscenza panoramica dei vari paradigmi di programmazione. Conoscenza del paradigma orientato agli oggetti. Conoscenza del linguaggio Java. Capacita' di applicare conoscenza e comprensione Capacita' di valutare le funzionalita' dei diversi paradigmi di programmazione. Capacita' di scrivere un codice nel linguaggio di programmazione Java. Autonomia di giudizio Capacita' di valutare e comparare autonomamente le soluzioni di un problema di limitata complessita'. Abilita' comunicative Capacita' di organizzarsi in gruppi di lavoro. Capacita' di comunicare efficacemente in forma orale, anche utilizzando termini in inglese. Capacita' di apprendere Capacita' di catalogare, schematizzare e rielaborare le nozioni acquisite.
VALUTAZIONE DELL'APPRENDIMENTO	L'esame finale consiste in una prova scritta e una prova orale. La prova scritta e' costituita da un test a risposta multipla di tipo vero/falso (5 opzioni di risposta per ogni domanda, una sola risposta esatta), costituito da 30 domande. Ogni risposta esatta vale un punto, ogni risposta sbagliata vale -0.25 punti, ogni risposta non fornita vale 0 punti. La prova orale consiste in un colloquio che verte ad approfondire le conoscenze acquisite e la capacita' di scrivere codice corretto in Java (su carta). Concorrono a determinare il voto: 1) Il risultato della prova scritta; 2) La chiarezza e correttezza delle risposte al colloquio orale; 3) La chiarezza e correttezza del codice Java scritto su carta durante il colloquio orale. Fasce di valutazione: Il voto va da 18/30 (prova scritta sufficiente, conoscenze minime sul programma, valutate in base alle risposte fornite alle domande rivolte al colloquio orale, capacita' di scrivere codice Java elementare) a 30/30, con eventuale lode (prova scritta eccellente, ottime conoscenze sul programma, valutate in base alle risposte fornite alle domande rivolte al colloquio orale, ottime capacita' di scrivere codice Java anche complesso).
OBIETTIVI FORMATIVI	Fornire le competenze di base, sia metodologiche che tecniche, sui linguaggi di programmazione, in particolar modo sui linguaggi orientati agli oggetti.
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali in laboratorio. Gli studenti avranno a disposizione dei computer per seguire le lezioni in modo interattivo. La frequenza delle lezioni e' fortemente raccomandata.
TESTI CONSIGLIATI	1) P. Principe, Java 9 Guida completa, 2017 Apogeo (per la parte su Java - for the part on Java); 2) C. Horstmann, Concetti di informatica e fondamenti di Java, VI ed. 2016, Apogeo; (per la parte su Java - for the part on Java); 3) M. Gabbrielli, S. Martini, Linguaggi di programmazione, principi e paradigmi, II ed. 2011, Mc GrawHill (per la parte Teoria dei Linguaggi di Programmazione - for the part Theory of Programming Languages)

PROGRAMMA

ORE	Lezioni
2	Introduzione e Primi Passi in Java: Sintassi di base - Primo programma - Commenti al codice - Letterali - Dichiarazione di variabili - Identificatori - Tipi primitivi: byte, short, int, long, float, double, boolean, char - Operatori unari, binari, condizionali e di concatenazione
3	Le classi Java I: Classi e oggetti - Attributi e metodi - Convenzioni su maiuscole e minuscole - Il metodo main - Costruttori - L'operatore new - Il costruttore di default - Riferimenti - Confronto di riferimenti - Variabili di istanza e variabili locali - Accesso ad attributi e metodi delle classi con dot notation - La Javadoc.
3	Le classi Java II: I metodi print, println e printf - Incapsulamento dei dati - Interfaccia pubblica e interfaccia privata - Modificatori di visibilita' public e private - Il riferimento implicito this - Deallocazione automatica della memoria e Garbage Collector - Firma di un metodo - Overloading di metodi - Overloading di costruttori - Invocazione di altri costruttori mediante "this()" - Variabili final
2	Le classi Java III: Attributi static - Metodi static - Organizzazione e compilazione di un package - Visibilita' package-private
2	Controllo del flusso: Costrutti di selezione, iterazione e salto: if, switch, while, do-while, for, break, continue, return - Il for esteso
6	Stringhe e Wrapper: Il tipo String - Istanziamento di stringhe - Il pool di stringhe - La codifica Unicode per i caratteri - L'operatore di concatenazione - Il metodo equals per le stringhe - Il metodo compareTo per le stringhe - I metodi per le stringhe: startsWith e endsWith, length, substring, contains - toLowerCase e toUpperCase, trim, charAt, indexOf, lastIndexOf, replaceFirst, replaceAll, valueOf - Espressioni regolari - I metodi che usano le espressioni regolari: matches, replaceFirst, replaceAll, split - La classe StringBuilder - I metodi append, insert, delete, deleteCharAt, replace, reverse - La classe Scanner e i suoi metodi: next, nextLine, nextInt, nextDouble - Le classi wrapper - Boxing e unboxing - I metodi isDigit e isLetter della classe Character - I metodi parseInt, parseShort, parseFloat - I metodi toBinaryString, toOctalString e toHexString

PROGRAMMA

ORE	Lezioni
3	Arrays e ArrayList: Creazione e inizializzazione di un array - Array di riferimenti a oggetti - L'attributo length - Il for esteso sugli array - I metodi della classe Arrays: equals, fill, sort, binarySearch - La classe ArrayList - Cenni su Generics e classi parametriche - I metodi per gli ArrayList: add, size, get, set, remove, indexOf, lastIndexOf, contains, toArray - I metodi della classe Collections: reverse, sort, max e min, shuffle, swap, binarySearch
3	Metodi Universali: I metodi presenti in tutte le classi - I metodi equals, toString e hashCode - Overriding del metodo toString - Overriding del metodo equals - Funzioni hash - Collisioni e hashing perfetto - Bucket e tabelle hash - Overriding del metodo hashCode - Mutua compatibilita' di questi overriding.
3	Ereditarieta' I: Riutilizzo e distribuzione dei programmi Java - Sottoclassi e superclassi - La parola chiave extends - Ereditarieta' di attributi e metodi - Il modificatore protected - Overriding di metodi - L'annotazione @Override - Attributi adombrati (shadowing) - Classi e metodi final
3	Ereditarieta' II: Ereditarieta' e costruttori - Invocare costruttori della superclasse con "super()" - Il riferimento super - Ereditarieta' e membri static - Ereditarieta' singola e Diamond Problem - Subtyping
3	Polimorfismo: Type checking statico - Binding dinamico - Definizione di polimorfismo - polimorfismo per metodi e polimorfismo per dati - array polimorfi - metodi virtuali e attributi adombrati - Il costrutto instanceof
5	Classi astratte e interfacce: Il modificatore abstract - Classi astratte e sottoclassi concrete - Classi astratte e membri static - Definizione e dichiarazione di un'interfaccia - Contenuto di un'interfaccia - Implementare un'interfaccia - Conflitto di nomi di variabili tra un'interfaccia e una classe che la implementa - Implementazioni parziali di interfacce in classi astratte - Estensione di interfacce - Ereditarieta' multipla per interfacce - Diamond problem, shadowing, overriding e overloading nell'ereditarieta' multipla per interfacce - Riferimenti a un'interfaccia - Interfacce ed ereditarieta' - Interfacce e polimorfismo - Focus sull'interfaccia Comparable - Comparatori e interfaccia Comparator - Implementazione di comparatori - Classi interne e classi interne anonime - Implementazione di un'interfaccia in una classe interna anonima
6	Input/output: Flussi (stream) di dati - Dati in ingresso e dati in uscita - Flussi di caratteri e flussi di byte - La classe File - I metodi della classe File: delete, exists, getAbsolutePath, getName, getPath, isDirectory, isFile, length, list, mkdir e renameTo - Usare Scanner per leggere dati da un file testuale - I metodi hasNext e hasNextLine, next e nextLine - Gestione di flussi di byte: le classi astratte InputStream e OutputStream e le loro sottoclassi FileInputStream e FileOutputStream - I metodi read e write per FileInputStream e FileOutputStream - Gestione di flussi di caratteri: le classi astratte Reader e Writer e le loro sottoclassi - I metodi read e write per FileReader e FileWriter - Bufferizzazione: le classi BufferedReader e BufferedWriter - La classe PrintWriter Il metodo printf - Filtraggio di classi - Flussi di oggetti - Le classi ObjectInputStream e ObjectOutputStream - Serializzazione e deserializzazione - L'interfaccia Serializable - I metodi readObject e writeObject - Serializzazione e membri static - Il modificatore transient - Memorizzazione dei dati di un programma su file
3	Eccezioni: Condizioni di errore in un programma - La classe Throwable e le sue sottoclassi Exception ed Error - Eccezioni controllate e non controllate - Gerarchia di eccezioni e subtyping - Il delegation model - Catturare o propagare un'eccezione - La clausola throws - Lancio manuale di un'eccezione mediante la parola chiave throw - Creazione di nuove classi di eccezioni - Propagazione di un'eccezione - Lo stack delle chiamate dei metodi e il System Exception Handler - Gestione delle eccezioni - Il costrutto try/catch - La clausola finally - Annotazioni Javadoc per le eccezioni
5	Strutture dati I: Programmazione generica in Java - Uso dei Generics - Wildcard e metodi generici - Strutture dati astratte e loro implementazioni - Modellizzazione di SDA attraverso interfacce - Panoramica del Java Collection Framework - L'interfaccia Collection - L'interfaccia Set - La classe HashSet - Capacity e load factor - L'interfaccia SortedSet - La classe TreeSet - Iteratori - Interfacce Iterable e Iterator - Differenze d'uso tra iteratori e for esteso - L'interfaccia List - L'interfaccia ListIterator - La classe ArrayList - La classe LinkedList - La SDA mappa (array associativo) - L'interfaccia Map - La sottointerfaccia Map.Entry - La classe HashMap - L'interfaccia SortedMap - La classe TreeMap - Ordinare gli elementi di una mappa per valore
2	Strutture dati II: Le SDA pila e coda - Scrittura delle interfacce Stack e Queue - Implementazione attraverso ArrayList e LinkedList - Differenza tra implementazione per estensione e per inclusione - Implementazione attraverso array - La tecnica del doubling-halving
3	Interfacce grafiche I: Introduzione a AWT e Swing - Struttura di una finestra - Il Pannello dei contenuti - la barra dei menu - Costruzione di una finestra - I layout e i layout manager - Gestione degli eventi - Panoramica sui principali ascoltatori - Le classi adapter - Implementazione di un ascoltatore come classe esterna, interna o anonima - L'interfaccia MouseListener e la classe MouseEvent - L'interfaccia ActionListener e la classe ActionEvent - L'interfaccia KeyListener e la classe KeyEvent - La classe Dimension.
3	Interfacce grafiche II: Menu, sottomenu e voci di menu - Toolbar - L'interfaccia Icon e la classe ImageIcon - La classe BorderLayout - I tool tip - La classe JLabel - La classe JButton - La gerarchia di JTextComponent - La classe JTextField - L'interfaccia DocumentListener e la classe DocumentEvent - La classe JCheckBox - L'interfaccia ItemListener e la classe ItemEvent - La classe JRadioButton - La classe JTextArea - La classe JScrollPane
6	Teoria dei linguaggi di programmazione I: Macchine astratte - Memoria e Interpretare - Operazioni dell'interprete - Ciclo Fetch/Execute dell'interprete - Linguaggio macchina di una macchina astratta - Esempi di macchine astratte: ristorante e macchina hardware - Implementazioni di una macchina astratta: realizzazione in hardware, emulazione mediante firmware, simulazione mediante software - Implementazione software e macchina ospite - Implementazione interpretativa pura e compilativa pura - Definizione formale di interprete - Definizione formale di compilatore - Vantaggi e svantaggi delle due implementazioni pure - Implementazioni ibride - La Java Virtual Machine - Macchine intermedie e gerarchie di macchine astratte.

PROGRAMMA

ORE	Lezioni
6	Teoria dei linguaggi di programmazione II: Nomi e oggetti denotabili - Importanza e uso dei nomi - Definizione di binding - Fasi di binding - Binding statici e binding dinamici - Tempo di vita di un binding e tempo di vita di un oggetto - Definizione di ambiente - Dichiarazioni - Aliasing - Blocchi - Annidamento dei blocchi - Differenza tra annidamento statico e annidamento dinamico - Visibilita' di una variabile - Regola di visibilita' canonica - Ambiente locale, ambiente non locale e ambiente globale - Visibilita' dall'interno verso l'esterno e dall'esterno verso l'interno - Buchi di visibilita' - Operazioni sui nomi e sull'ambiente - Regole di scope statico e di scope dinamico.