

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

Id	Quesito	Risp_esatta	Risp_errata_1	Risp_errata_2	Materia
1	Nello sviluppo di un programma, quale tra le seguenti fasi viene effettuata prima delle altre?	Codifica	Debugging	Testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
2	Cosa è un algoritmo?	Un processo passo, passo per risolvere un problema.	Un modello di programma.	Un software per risolvere problemi matematici.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
3	Quali delle seguenti voci fanno parte dei processi di Ingegneria del Software:	Analisi, progettazione, sviluppo, rilascio, manutenzione evolutiva	Analisi, produzione, misurazione, risk management	Modellazione, sviluppo, data analysis, testing, backup	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
4	Quale tra le seguenti è una caratteristica di un buon progetto software?	Realizzare tutti i requisiti descritti nell'analysis model.	Avere un accoppiamento forte tra i moduli	Includere i test per tutte le componenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
5	Quale tra le seguenti NON è una delle attività quadro previste dall'ingegneria del software	Marketing	Pianificazione	Modellizzazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
6	Quale, tra le seguenti fasi del ciclo di vita del software, avviene dopo tutte le altre indicate?	Manutenzione	Progettazione	Analisi dei requisiti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
7	Nel controllo di qualità del software, quando viene eseguito il processo chiamato "validazione"?	Al termine dello sviluppo	Dopo ogni fase del ciclo di vita	Dopo la definizione dei requisiti per verificare con il cliente che nessun requisito fondamentale sia stato omesso	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
8	Cos'è l'HTML?	Un linguaggio di scripting	Un linguaggio di programmazione	Un protocollo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
9	Quale dei seguenti non è un linguaggio di scripting?	Postscript	XML	Javascript	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
10	Vengono utilizzati per identificare un utente che ritorna su un sito web	Cookies	Plug-ins	Scripts	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
11	Cosa descrive il piano di manutenzione?	Le procedure di manutenzione	Le procedure di controllo qualità	Le procedure di configurazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
12	Cosa si intende per 'usabilità' di un software?	E' la capacità del prodotto software di essere capito, appreso, usato e benaccetto dall'utente.	E' la facilità con la quale è possibile analizzare il codice per localizzare un errore nello stesso.	E' la capacità di utilizzo di quantità e tipo di risorse in maniera adeguata.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
13	Cosa si intende per 'efficienza' di un software?	E' la capacità di fornire appropriate prestazioni relativamente alla quantità di risorse usate.	E' la capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso.	E' la capacità di mettere in condizione gli utenti di farne uso per i propri scopi e controllarne l'uso.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
14	Cosa si intende per 'manutenibilità' di un software?	E' la capacità del software di essere modificato, includendo correzioni, miglioramenti o adattamenti.	E' la capacità di fornire adeguati tempi di risposta.	E' la capacità del software di essere installato in uno specificato ambiente.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
15	I programmi per computer sono scritti in un linguaggio di programmazione di alto livello;come viene definita la versione leggibile dall'uomo	Codice sorgente	Cache	Set di istruzioni	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
16	Un insieme di procedure passo passo per portare a termine un compito è noto come:	algoritmo	firmware	programma di sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
17	La fase iniziale del ciclo di vita del software prevede:	Studio di fattibilità e Analisi dei requisiti	Progettazione del software	Individuazione dell'ambiente di sviluppo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
18	Lo studio di fattibilità:	E' un analisi preliminare che valuta la fattibilità di un progetto, i costi e l'impatto sull'organizzazione che dovrebbe utilizzarlo	Valuta esclusivamente i costi del progetto e la disponibilità economica del committente	Dovrebbe valutare l'infrastruttura hardware e software che costituirà l'ossatura del progetto	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

19	La fase di analisi dei requisiti viene fatta:	Tramite interviste, osservazione diretta, focus group	Tramite lo studio di realtà applicative simili	E' una fase trascurabile della progettazione che può essere omessa	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
20	La programmazione strutturata si basa sull'utilizzo di 3 costrutti sintattici fondamentali. Quali?	Sequenza - selezione - iterazione	Selezione - profanazione - iterazione	Iterazione - sequenza - analisi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
21	La metodologia di programmazione strutturata considera un algoritmo come un insieme di blocchi di istruzioni ognuno fornito di un solo ingresso e una sola uscita . Ciascun blocco di istruzioni è:	isolato dagli altri	collegato agli altri	conforme gli altri	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
22	Il metodo di ordinamento bubble sort o “a bolle” consiste nel confronto degli elementi :	a due a due	a uno a uno	a tre a tre	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
23	L'ordinamento per scambio è detto anche :	ingenuo	imprevisto	iterativo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
24	In cosa consiste la tecnica di ordinamento con metodo Shell?	Nell'utilizzo di tre cicli <i>while</i> ed un ciclo <i>for</i>	Nell'utilizzo di due cicli <i>while</i> ed un ciclo <i>for</i>	Nell'utilizzo di un ciclo <i>while</i> ed un ciclo <i>for</i>	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
25	In generale un algoritmo di ricerca si pone come obiettivo quello di:	trovare, all'interno di un gruppo, un elemento avente determinate caratteristiche.	eliminare le parti di codice errato	omogenizzare i dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
26	Dato un problema, esistono diversi algoritmi che permettono di risolverlo. I fattori che possono influenzare la scelta dell'algoritmo migliore sono principalmente due:	Tempo di elaborazione e quantità di memoria	Capacità della memoria e velocità di interpretazione	Quantità e qualità dei codici	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
27	Quali sono le figure generalmente coinvolte nell'analisi preliminare?	Responsabile del progetto, Analista, Esperto del contesto di Business	Vertici aziendali, progettista, rivenditore prodotti software	Utenti finali	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
28	L'analisi preliminare deve produrre:	Documento di analisi che contiene le specifiche di cosa il software dovrà fare e le risorse umane necessarie	Documento di analisi che contiene le specifiche in termini di hardware da acquistare	Il documento di progetto con le specifiche in termini di linguaggi di programmazione e DBMS utilizzati, le specifiche funzionali e le specifiche sui dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
29	Il class diagram del linguaggio UML;	Rappresenta le classi come box contenenti gli attributi della classe e collegate fra loro da linee che rappresentano i diversi tipi di relazione esistenti tra le classi	E' equivalente al modello Entità Relazione	Rappresenta gli utenti del sistema da realizzare secondo un modello gerarchico	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
30	Nella progettazione il Data Flow Diagram:	E' una rappresentazione grafica del flusso delle informazioni tra i diversi moduli di un sistema informatico	E' un formalismo grafico per definire in modo astratto i dati utilizzati da un sistema	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
31	Cosa si intende in generale per paradigmi di programmazione?	Una collezione di modelli concettuali che insieme plasmano il processo di analisi, progettazione e sviluppo di un programma	Si tratta di uno o più specifici linguaggi di programmazione orientati agli oggetti	E' una famiglia di linguaggi di programmazione sviluppati ed in uso prima dell'anno 2000	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
32	Per "programmazione procedurale" di intende:	E' un paradigma di programmazione che consiste nel creare dei blocchi di codice sorgente, identificati da un nome e racchiusi da dei delimitatori	E' un paradigma di programmazione che consiste nel creare degli oggetti e delle istanze di essi	E' una caratteristica dei DBMS Relazionali e corrisponde alla caratteristica di suddividere i dati in tabelle e relazioni	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
33	Come può essere definito il collaudo?	Il <i>controllo</i> definitivo a che un prodotto software finito rispetti i <i>requisiti</i> del <i>committente</i>	Un primo livello di controllo circa il funzionamento del software	Una fase secondaria della validazione software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

34	Quale delle seguenti affermazioni, relative alla tecnica di test “white box”, è corretta?	Include il test dei loop	È solitamente effettuata nella fase di integrazione	È solitamente effettuata dopo il “black box” testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
35	La "schedulazione dei test" è:	una lista di attività, compiti o eventi del processo di test, che identifica le date e/o i tempi di inizio e fine previsti, e le loro interdipendenze	la documentazione che riassume le attività e i risultati dei test	un tipo di test in cui le test suite sono eseguite su dispositivi fisici o virtuali	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
36	Un bug è:	un errore nel software	un'area della memoria RAM	un dispositivo hardware	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
37	Di quale fase del ciclo di vita di un sistema informativo sono tipiche le regolari operazioni di salvataggio e monitoraggio?	Gestione	Realizzazione	Progettazione e pianificazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
38	Nel controllo di qualità del software, quando viene eseguito il processo chiamato "validazione"?	Al termine dello sviluppo	Dopo ogni fase del ciclo di vita	Dopo ogni richiesta di modifica; la “validazione” è infatti il processo con cui si decide se l’intervento di manutenzione richiesto sia veramente necessario	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
39	Uno solo dei seguenti linguaggi è Event driven, quale?	Visual Basic	Assembler	C++	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
40	L'affidabilità può essere definita come la caratteristica di qualità di un prodotto software relativa alla sua:	capacità di funzionare correttamente nel tempo	possibilità di essere impiegato produttivamente da una particolare utenza	capacità di superare il primo test di funzionamento	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
41	L'accettazione rappresenta:	l’ultima fase del ciclo di vita in cui il prodotto è sottoposto alla valutazione del committente	un ambiente controllato che permette il <i>controllo dinamico</i> di un <i>modulo</i> o di un <i>sistema software</i> indipendentemente da fattori esterni che potrebbero influenzarne il comportamento e quindi inquinare i risultati del <i>test</i>	una tecnica di <i>integrazione</i> dei <i>moduli</i> che consiste nel provare il <i>sistema software</i> completo immediatamente dopo aver controllato i singoli moduli	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
42	Quale delle seguenti affermazioni sul linguaggio Javascript è vera?	È un linguaggio di programmazione che viene interpretato da un interprete incorporato in un browser web	È un’evoluzione del linguaggio di programmazione Java	È una versione di HTML dinamico	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
43	Quale tra le seguenti non è una caratteristica del software?	Competitività	Manutenibilità	Affidabilità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
44	Nella programmazione strutturata è sconsigliato l’uso	del salto incondizionato	degli annidamenti di istruzioni	della ricorsione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
45	Nel passaggio dei parametri per valore, il parametro a run time è:	Una costante	Una funzione	Una variabile	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
46	Nella programmazione ad oggetti, come è possibile creare nuove classi estendendo classi già esistenti?	Mediante l’ereditarietà	Mediante l’incapsulamento	Mediante gli attributi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
47	Una variabile globale è:	una variabile visibile anche in ogni sotto procedura del programma principale	una variabile passata come parametro ad una procedura	un’espressione booleana	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
48	Quale tra i seguenti non è un concetto ammesso nella programmazione strutturata?	Salto incondizionato	Iterazione	Selezione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

49	In un algoritmo di ricerca sequenziale:	La ricerca viene effettuata tramite una serie di confronti tra il valore da ricercare e tutti gli elementi della struttura dati	La ricerca viene preceduta da un ordinamento degli elementi della struttura dati	La ricerca avviene casualmente all'interno della struttura dati scartando, ad ogni nuova ricerca, gli elementi che non corrispondono al criterio di ricerca	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
50	Secondo l'algoritmo di ricerca binaria:	La struttura dati che contiene il set di valori su cui effettuare la ricerca deve essere preventivamente ordinata	La struttura dati che contiene il set di valori su cui effettuare la ricerca non deve essere preventivamente ordinata	La ricerca viene effettuata tramite una serie di confronti tra il valore da ricercare e tutti gli elementi della struttura dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
51	Cosa si intende per "Portabilità" di un linguaggio di programmazione?	L'indipendenza dalla macchina fisica, ovvero la possibilità di eseguire un programma su una diversa piattaforma	La facilità di comprendere finalità e funzionamento di un programma esaminando il codice sorgente	E' la misura dell'efficienze del linguaggio rispetto al problema da risolvere	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
52	Cosa si intende per "Affidabilità" del Software?	Un software si può dire affidabile quando produce risultati corretti in base all'input ricevuto	Un software si può dire affidabile quando si avvia sempre correttamente senza rilasciare errori	Un software si può dire affidabile quando esegue velocemente il suo compito	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
53	Cosa si intende per "Robustezza" del Software?	E' la capacità di un software di comportarsi in maniera ragionevole in caso si trovi in situazioni non previste dalle specifiche	E' la capacità di un software di lavorare per lunghi periodi mantenendo sempre le stesse prestazioni	E' la capacità di un software di continuare a lavorare anche a fronte di problemi all'hardware che lo ospita	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
54	Cosa si intende per "Usabilità" del Software?	Un software si dice usabile quando gli utenti ne trovano facile ed intuitivo l'utilizzo.	Un software si dice usabile quando produce risultati corretti in base all'input ricevuto	Un software si dice usabile quanto può essere utilizzato su diversi sistemi operativi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
55	Cosa si intende per "Riusabilità" del Software?	La possibilità di riutilizzare un prodotto o parti di esso (magari anche con qualche modifica) per costruire un prodotto nuovo o diverso	Un software si dice usabile quando gli utenti ne trovano facile ed intuitivo l'utilizzo.	Un software si dice usabile quanto può essere utilizzato su diversi sistemi operativi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
56	Secondo il "modello classico del ciclo di vita del software" quale tra le seguenti affermazioni è vera:	L'analisi dei requisiti precede la fase di progettazione	L'analisi dei requisiti avviene immediatamente prima della fase di sviluppo e test	L'analisi dei requisiti precede lo studio di fattibilità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
57	Secondo il "modello classico del ciclo di vita del software" quale tra le seguenti affermazioni è vera:	Lo studio di fattibilità è sempre la prima fase di un qualsiasi progetto	Lo studio di fattibilità può essere fatto dopo l'analisi dei requisiti	Lo studio di fattibilità può essere fatto dopo il rilascio del prodotto finale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
58	Secondo il "modello classico del ciclo di vita del software" quale tra le seguenti affermazioni è vera:	La manutenzione è l'ultima fase del ciclo di vita del software	La manutenzione non fa parte del ciclo di vita del software	La manutenzione viene eseguita immediatamente dopo la fase di sviluppo e test del sistema e immediatamente prima dell'installazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
59	Secondo il "modello classico del ciclo di vita del software" quale tra le seguenti affermazioni è vera:	La fase di sviluppo e test del sistema viene eseguita dopo la fase di progettazione	La fase di sviluppo e test del sistema viene eseguita immediatamente prima della fase di progettazione	La fase di sviluppo e test del sistema viene eseguita immediatamente prima dell'analisi dei requisiti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
60	Secondo il "modello classico del ciclo di vita del software" quale tra le seguenti affermazioni è vera:	La fase di installazione segue quella Integrazione e Test di sistema	La fase di installazione è immediatamente successiva a quella di sviluppo	La fase di installazione può essere omessa	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
61	I modelli di sviluppo del software a "processo evolutivo":	Prevedono un ciclo di sviluppo in cui un prototipo iniziale evolve gradualmente verso il prodotto finito	Sono equivalenti al modello di sviluppo classico	Prevedono che il prodotto finale sia completato in ogni sua parte prima della consegna ed installazione presso il committente	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
62	Cosa si intende per "programmazione ad eventi"?	E' una tecnica di programmazione in cui il flusso del programma è largamente determinato dal verificarsi di eventi esterni	E' una tecnica di programmazione in cui l'esecuzione delle istruzioni segue percorsi fissi che si determinano in punti del programma definiti dal programmatore	E' la gestione di eventi imprevisti che possono capitare durante il normale funzionamento di un programma	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

63	Cosa si intende per "programmazione modulare"?	La struttura di un programma è divisa in moduli, ovvero blocchi di codice che svolgono intere funzioni all'interno del programma	E' un paradigma di programmazione in cui il flusso di esecuzione di un programma assume la forma di una serie di valutazioni di funzioni matematiche	E' una tecnica di programmazione in cui il flusso del programma è largamente determinato dal verificarsi di eventi esterni	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
64	Cosa si intende per "Malfunzionamento" di un Software?	E' un funzionamento di un programma diverso da quanto previsto dalla sua specifica	Quando il software viene attaccato da un hacker e ne modifica il comportamento	Quando un software non è compatibile con uno specifico sistema operativo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
65	Il collaudo formale di un software in caso di una gara d'appalto di una PA viene eseguito da:	Personale tecnico ed amministrativo della PA e personale della ditta che ha sviluppato il software	Esclusivamente personale tecnico ed amministrativo della PA che dovrà poi utilizzare il prodotto	Esclusivamente personale tecnico della ditta che ha sviluppato il prodotto	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
66	Cosa si intende per "Ambiente di test"?	E' un ambiente controllato, simile a quello di produzione, in cui poter eseguire dei test su un software	E' l'ambiente hardware e software di produzione dove sarà collaudato il software	E' un ambiente, il più dissimile da quello di produzione in cui poter eseguire dei test su un software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
67	Cosa si intende per "Debugging"?	E' quell'attività che ha lo scopo di eliminare gli errori da un software attraverso una fase di test da effettuarsi prima della messa in esercizio	E' quell'attività in cui si effettua la formazione al personale che dovrà utilizzare un software	E' quell'attività in cui vengono introdotte	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
68	Cosa si intende per "Stress Test" di un Software?	E' un'attività di test del software che determina la solidità dello stesso testandolo oltre i limiti del normale funzionamento	E' un tipo di test del software che viene eseguito su un pc su cui è in esecuzione uno stress test hardware	E' un tipo di test in cui si spegne/accende in continuazione un pc, verificando che il software su questo in esecuzione continui a lavorare normalmente	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
69	L'attività di "security test" nella fase di debug di un software prevede:	Un controllo sull'accesso ai dati o a funzionalità che dovrebbero essere riservate, si controlla l'efficacia dei meccanismi di sicurezza del sistema	E' un controllo sul server che ospita il software per verificare che non abbia falle di sicurezza	Corrisponde all'hardening del sistema operativo su cui sarà eseguito il software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
70	Cosa si intende per "difetto quiescente" di un software?	E' un difetto del software che non si manifesta per un lungo periodo perché nascosto in parti del programma raramente utilizzate	E' un difetto del software che si manifesta con cadenza periodica es. una volta alla settimana o ad un orario ben preciso	E' un difetto che si manifesta sulla stessa procedura testata solo a determinati utenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
71	Gli array sono una struttura dati:	Multidimensionale	Esclusivamente unidimensionale	Esclusivamente bidimensionale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
72	Come si definisce un "array"?	Un array può essere definito come una “collezione organizzata di oggetti”	Un array può essere definito come una cella di memoria atta a contenere una informazione	Un array può essere definito come un file fisico memorizzato sul file system	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
73	Per identificare un dato elemento di un array è necessario utilizzare oltre al nome dell'array anche un numero indice che serve per:	specificare la posizione dell'elemento che si sta cercando all'interno dell'array	Definire il numero massimo di elementi di cui è costituito l'array	Definire la posizione dell'array nel file system del computer	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
74	Nelle istruzioni "iterative" :	Un determinato blocco di codice viene eseguito fino a che risulta soddisfatta una determinata condizione	Non possono essere contenute delle istruzioni condizionali	Un determinato blocco di codice viene eseguito in alternativa ad un altro	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
75	Nelle istruzioni "condizionali" :	Un determinato blocco di codice viene eseguito in alternativa ad un altro	Non possono essere contenute delle istruzioni condizionali nidificate	Un determinato blocco di codice viene eseguito fino a che risulta soddisfatta una determinata condizione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
76	Nel linguaggio di programmazione python un array può essere formato da elementi di tipi diversi tra loro?	Si	No	Si, ma solo se si sviluppa in ambiente Linux	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

77	Se in una istruzione iterativa la condizione che ne determina la fine non cambia mai di stato:	Il ciclo non termina mai	Manca una istruzione che termina il ciclo, l'interazione viene interrotta automaticamente dopo un tempo prefissato da parte del sistema operativo	Da un punto di vista del codice non è possibile definire un ciclo che sia senza fine	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
78	Quale tra le seguenti sono istruzioni iterative che si trovano nella quasi totalità dei linguaggi di programmazione?	While; For	If; Do	Case; Next	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
79	In un istruzione condizionale, in base al valore della condizione:	Se la condizione è vera viene eseguito un blocco di codice, se è falsa si passa all'istruzione successiva o si esegue un altro blocco di codice alternativo al primo	A prescindere dalla condizione vengono eseguiti tutti i blocchi di codice contenuti nel blocco di istruzioni che contiene la condizione	Se la condizione è falsa il programma termina	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
80	Nei linguaggi di programmazione le variabili "booleane" possono assumere valori:	True e False	A e B	100 e 001	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
81	Nell'ambito dei processi di sviluppo del software, cosa prevede la fase di analisi?	Lo studio del contesto nel quale il progetto software andrà ad inserirsi, delle caratteristiche o dei requisiti che deve esibire e degli eventuali costi ed aspetti logistici della sua realizzazione	La definizione delle linee essenziali della struttura del prodotto software in funzione ai requisiti evidenziati nell'attività di analisi, con lo scopo di definire la soluzione al problema in maniera più dettagliata	La verifica e la validazione del software, in modo tale che i requisiti individuati nell'analisi vengano soddisfatti. L'infrastruttura di supporto utilizzata in tale fase è detta ambiente di testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
82	Nell'ambito dei processi di sviluppo del software, cosa prevede la fase di progettazione?	La definizione delle linee essenziali della struttura del prodotto software in funzione ai requisiti evidenziati nell'attività di analisi, con lo scopo di definire la soluzione al problema in maniera più dettagliata	Lo studio del contesto nel quale il progetto software andrà ad inserirsi, delle caratteristiche o dei requisiti che deve esibire e degli eventuali costi ed aspetti logistici della sua realizzazione	La verifica e la validazione del software, in modo tale che i requisiti individuati nell'analisi vengano soddisfatti. L'infrastruttura di supporto utilizzata in tale fase è detta ambiente di testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
83	Nell'ambito dei processi di sviluppo del software, cosa prevede la fase di collaudo o testing?	La verifica e la validazione del software, in modo tale che i requisiti individuati nell'analisi vengano soddisfatti. L'infrastruttura di supporto utilizzata in tale fase è detta ambiente di testing	La definizione delle linee essenziali della struttura del prodotto software in funzione ai requisiti evidenziati nell'attività di analisi, con lo scopo di definire la soluzione al problema in maniera più dettagliata	Lo studio del contesto nel quale il progetto software andrà ad inserirsi, delle caratteristiche o dei requisiti che deve esibire e degli eventuali costi ed aspetti logistici della sua realizzazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
84	Nell'ambito dei processi di sviluppo del software, cosa prevede la fase di pubblicazione e messa in opera?	La consegna della versione definitiva del prodotto software dopo aver superato con successo le verifiche della fase di collaudo	La definizione delle linee essenziali della struttura del prodotto software in funzione ai requisiti evidenziati nell'attività di analisi, con lo scopo di definire la soluzione al problema in maniera più dettagliata	Lo studio del contesto nel quale il progetto software andrà ad inserirsi, delle caratteristiche o dei requisiti che deve esibire e degli eventuali costi ed aspetti logistici della sua realizzazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
85	Nell'ambito dei processi di sviluppo del software, cosa prevede la fase di manutenzione?	Lo svolgimento di operazioni di correzione del prodotto software successive al rilascio, al fine di correggere ulteriori errori	La consegna della versione definitiva del prodotto software dopo aver superato con successo le verifiche della fase di collaudo	Lo studio del contesto nel quale il progetto software andrà ad inserirsi, delle caratteristiche o dei requisiti che deve esibire e degli eventuali costi ed aspetti logistici della sua realizzazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
86	Nell'ambito dei processi di sviluppo del software, il rilascio di patch cosa rappresenta?	Una manutenzione correttiva	Una manutenzione adattiva	Una manutenzione evolutiva	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
87	Nell'ambito dei processi di sviluppo del software, l'estensione delle funzionalità cosa rappresenta?	Una manutenzione evolutiva	Una manutenzione correttiva	Una manutenzione adattiva	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
88	Nell'ambito dei processi di sviluppo del software, l'adattamento a nuovi ambienti operativi è una:	manutenzione adattiva	manutenzione evolutiva	manutenzione correttiva	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

89	Il primo modello teorizzato del processo di sviluppo software è la metodologia a cascata (Waterfall) che prevede l'esecuzione:	sequenziale delle varie attività	non sequenziale delle varie attività	alternata delle varie attività	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
90	Che cosa è ActiveX?	Un linguaggio di programmazione sviluppato dalla Microsoft per costruire applicazioni web integrate con ambienti Windows	Un linguaggio di programmazione per i Videofonini	Un sistema operativo utilizzato per costruire applicazioni web integrate con ambienti Windows	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
91	Uno dei principali vantaggi della tecnica di programmazione orientata agli oggetti (OOP) rispetto alla procedurale è che consente ai programmatori di creare _____ che non hanno bisogno essere modificati quando viene aggiunto un nuovo tipo di oggetto:	Moduli	Strutture Dati	Record	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
92	Cloud computing riguarda il modo in cui un software viene:	Distribuito	Cancellato	Definito	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
93	Quali errori possono essere segnalati dal compilatore?	Errori di sintassi	Errori di logica	Errori di runtime	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
94	Con riferimento allo sviluppo del software Object-Oriented, gli 'oggetti' ...	Incapsulano sia i dati che le funzioni che li manipolano	Sono descrizioni generalizzate che descrivono un insieme di oggetti simili	Sono i mezzi con cui vengono scambiati messaggi all'interno di un sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
95	Con riferimento allo sviluppo del software Object-Oriented, le 'classi'...	Sono descrizioni generalizzate che descrivono un insieme di oggetti simili	Incapsulano sia i dati che le funzioni che li manipolano	Sono i mezzi con cui vengono scambiati messaggi all'interno di un sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
96	Con riferimento allo sviluppo del software Object-Oriented, la gerarchia della classe	È il modo con cui gli “attributi” e i “metodi” di una superclasse sono “ereditati” dalle sue sottoclassi	È una collezione di oggetti	È un meccanismo per diffondere i cambiamenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
97	Con riferimento allo sviluppo del software Object-Oriented, cos'è l'ereditarietà?	È un modo per consentire alle sottoclassi di riutilizzare i dati e le procedure di superclassi	Sono descrizioni generalizzate che descrivono un insieme di oggetti simili	È una collezione di oggetti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
98	Secondo le prescrizioni del controllo di qualità di un prodotto software, ad una data fase del processo di sviluppo, cosa viene verificato durante l'audit?	Che le specifiche della fase precedente siano soddisfatte	Che il cliente sia stato informato degli errori eliminati rispetto alla fase precedente	Che il prodotto sia conforme alle specifiche iniziali	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
99	A che cosa serve il testing di regressione (regression testing)?	A verificare che non siano stati introdotti errori in parti di software già testate	A garantire che sia possibile ripristinare la precedente versione installata	A verificare che esistano tutti i documenti previsti nel piano di qualità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
100	Con quale strumento si traduce un codice sorgente in codice oggetto?	Compilatore.	Interprete.	Editor.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
101	I diagrammi di controllo dei flussi (Control flow diagrams) sono...	Necessari per la modellizzazione di sistemi guidati da eventi (event driven systems)	Necessari per la modellizzazione di tutti i sistemi	Talvolta utilizzati al posto dei diagrammi di flusso dei dati (data flow diagrams)	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
102	Quale delle seguenti affermazioni relative al modello di sviluppo del software chiamato “evolutionary prototyping” è corretta?	L'obiettivo di tale modello di sviluppo è la consegna di un prodotto finito e funzionante all'utente finale	Nel modello di sviluppo del software chiamato “evolutionary prototyping” non si incomincia lo sviluppo finché i requisiti sono completamente identificati e descritti	L'obiettivo di tale modello di sviluppo è la definizione dei requisiti; lo sviluppo vero e proprio, normalmente, viene effettuato usando un altro modello	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
103	In informatica, qual è il significato più comune del termine "deployment"?	È la consegna al cliente, con relativa installazione e messa in funzione, di una applicazione o di un sistema software	l'assegnazione a ciascun gruppo di sviluppo di una delle parti in cui è stato suddiviso il progetto	È la fase del ciclo di vita di un prodotto software in cui il cliente accetta le fasi di sviluppo e la data di consegna prevista	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
104	Quale tra le seguenti NON è un'attività legata alla messa in servizio di un prodotto software?	Fatturazione delle attività	Organizzazione di un meccanismo per acquisire e gestire le valutazioni (feedback) degli utenti	Distribuzione dei supporti e della documentazione agli utenti finali	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

105	Quale tra le seguenti è un'attività legata alla messa in servizio di un prodotto software?	Distribuzione dei supporti e della documentazione agli utenti finali	Fatturazione delle attività	Integrazione dei componenti già collaudati nel prodotto finale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
106	Un diagramma di flusso di lavoro rappresenta:	Una sequenza di attività	Il ruolo degli utenti coinvolti	Il termine di un progetto	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
107	Il cloud computing ha cambiato il modo in cui il software viene...	distribuito	definito	cancellato	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
108	Differenza tra test Black-Box e White-Box.	Il test black-box che si concentra sulla funzionalità, l'obiettivo del metodo di test white-box è quello di eseguire l'analisi della struttura interna del software e della logica sottostante	Il test white-box separa il punto di vista degli utenti finali da quello degli sviluppatori, il test black-box viene solitamente eseguito da sviluppatori professionisti.	Con il test black-box i tester sono a conoscenza della struttura interna del codice sorgente del software che stanno testando, nei White Test no.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
109	Il Benchmarking:	è un processo di confronto delle prestazioni di un sistema con uno standard di settore	indica la velocità del processore	è una tecnica intesa a porre merci e servizi a disposizione del consumatore	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
110	Quale fase non è prevista nel ciclo di vita del software(SDLC)?	Astrazione	Progettazione	Implementazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
111	PHP è un linguaggio...	poco tipizzato	fortemente tipizzato	un linguaggio client	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
112	Cosa significa AJAX?	Asynchronous javascript and xml	Advanced JSP and xml	Asynchronous JSP and xml	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
113	Il seguente teorema da chi è stato enunciato ? “Un algoritmo scritto secondo le regole della programmazione a salti, per quanto complesso, può essere sempre trasformato in un algoritmo a esso equivalente che utilizzi esclusivamente tre costrutti sintattici fondamentali: sequenza, selezione e iterazione.”	Bohm e Jacopini	Fisherman e Artesi	Cluster e Boumer	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
114	Come è anche detto il metodo della ricerca binaria ?	Dicotomica	Distonica	Direzionale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
115	Il teorema fondamentale dell'algebra di Boole afferma che:	qualunque forma di elaborazione sui dati codificati in codice binario può ricondursi ad un'opportuna successione delle sole tre operazioni logiche fondamentali AND, OR e NOT che sono i passi elaborativi più elementari a livello macchina	qualunque forma di elaborazione sui dati codificati non in codice binario può ricondursi ad un'opportuna successione delle sole tre operazioni logiche fondamentali AND, OR e NOT che sono i passi elaborativi più elementari a livello macchina	Tutte le altre risposte sono esatte	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
116	Nella progettazione del software cosa si intende per "Requisiti Architettureali" ?	I prodotti e piattaforme che si devono usare, La struttura generale del sistema	Dati, performance, affidabilità	Cosa fa il sistema, come deve interagire con gli utenti o con altri sistemi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
117	Nella progettazione del software cosa si intende per "Requisiti Funzionali" ?	Cosa fa il sistema, come deve interagire con gli utenti o con altri sistemi	Sono quelli relativi alle prestazioni generali del sistema	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
118	Nella progettazione del software cosa si intende per "Requisiti di progetto" ?	Tempo, budget, risorse, logistica	I prodotti e piattaforme che si devono usare, La struttura generale del sistema	cosa fa il sistema, come deve interagire con gli utenti o con altri sistemi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
119	Nella progettazione del software cosa si intende per "Requisiti Legali" ?	Requisiti sulle normative a cui deve aderire il sistema o sull'uso di determinati protocolli previsti dalla legislazione	Requisiti relativi alla proprietà dei software e relative licenze d'uso	Sono quelli relativi alle prestazioni generali del sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

120	A cosa serve l'UML?	E' un formalismo grafico per descrivere e documentare l’architettura e le funzionalità di un sistema	E' un formalismo grafico per definire in modo astratto i dati utilizzati da un sistema	E' uno schema che descrive i costi di un progetto	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
121	Divide-et-impera:	E' una tecnica di programmazione che consiste nel partizionare un problema in sottoproblemi più piccoli dello stesso tipo; Risolvere i sottoproblemi; Ricombinare le soluzioni parziali per ottenere la soluzione del problema originale	E' un tipo di linguaggio di programmazione di basso livello che viene utilizzato direttamente a livello della Cpu	E' una particolare architettura software che prevede che ogni procedura sia distribuita su un diverso nodo della rete	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
122	Cosa si intende per Ereditarietà ?	L'ereditarietà è una relazione di generalizzazione/specializzazione: la superclasse definisce un concetto generale e la sottoclasse rappresenta una variante specifica di tale concetto generale	E' la caratteristica di alcuni linguaggi di programmazione di raccogliere in un unico oggetto logico i dati ed i metodi che agiscono su di essi	E' la possibilità di creare una ridondanza a livello funzionale in un applicazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
123	Quale tra le seguenti rappresenta una corretta definizione di "malfunzionamento o guasto (failure)"?	Si tratta di un funzionamento di un programma diverso da quanto previsto dalla sua specifica	Rappresenta l'origine del difetto	E' la causa del mancato funzionamento	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
124	Come può essere definito il walkthrough?	Una tecnica di analisi formale del codice che prevede due fasi: nella prima gli ispettori esaminano il codice simulandone l’esecuzione in assenza del gruppo che l’ha sviluppato, nella seconda fase i due gruppi discutono i problemi individuati	Una tecnica di analisi formale del codice che prevede due fasi: nella prima gli ispettori esaminano il codice simulandone l’esecuzione in presenza del gruppo che l’ha sviluppato, discutendo dei problemi individuati	La possibilità che un <i>programma</i> , in seguito a una modifica dovuta a un intervento di <i>manutenzione</i> o a un <i>processo</i> di sviluppo incrementale, peggiori le proprie funzionalità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
125	Individuare tra le seguenti opzioni di risposta la corretta definizione di "compatibilità".	Capacità di un componente o sistema di scambiare informazioni con altri componenti o sistemi, e/o di eseguire le sue funzioni richieste mentre condividono lo stesso ambiente hardware o software	Capacità di un componente o sistema di assicurare che i dati siano accessibili solo a coloro autorizzati ad avere accesso	Aderenza di un componente o sistema agli standard, convenzioni o norme legali e prescrizioni similari	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
126	Individuare tra le seguenti opzioni di risposta la corretta definizione di "installabilità".	Capacità di un componente o sistema di poter essere installato e/o disinstallato con successo in un specifico ambiente	Incapacità di un componente o sistema di poter essere installato e/o disinstallato con successo in un specifico ambiente	Capacità di un componente o sistema di scambiare informazioni con altri componenti o sistemi, e/o di eseguire le sue funzioni richieste mentre condividono lo stesso ambiente hardware o software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
127	La "scalabilità" di un prodotto software è:	la sua capacità di essere esteso (upgrade) per adattarsi a carichi crescenti	la sua incapacità di essere esteso (upgrade) per adattarsi a carichi crescenti	una lista di attività, compiti o eventi del processo di test, che identifica le date e/o i tempi di inizio e fine previsti, e le loro interdipendenze	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
128	In cosa consiste la prova formale?	Si tratta di una tecnica di controllo statico basata sulla dimostrazione dell’equivalenza fra un programma e la sua specifica funzionale	Rappresenta il contesto tipico di utilizzo di un’applicazione, dato dalle caratteristiche dell’ambiente operativo medio e dall’insieme di dati d’ingresso più probabili	Una “non tecnica” di <i>test</i>	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
129	Cosa si intende per UNICODE?	Lo standard internazionale di codifica dei caratteri dei linguaggi naturali	Un codice di programmazione univoco	Un software applicativo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
130	Un algoritmo euristico:	fornisce un’alternativa quando la risoluzione del problema ottimo può essere impossibile	costituisce la strategia ideale per risolvere problemi non troppo complessi	fornisce soluzioni molto costose in termini di tempo o di capacità di elaborazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
131	La complessità computazionale di un algoritmo è una misura:	del numero di istruzioni da eseguire per ottenere l’output	del numero di istruzioni che compongono l’algoritmo	della correttezza dell’algoritmo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

132	Quale delle seguenti affermazioni non si riferisce al paradigma di programmazione logica?	In PROLOG i fatti e le regole non possono avere variabili e costanti	Il PROLOG è l'esempio principale di linguaggio di programmazione basato sulla logica	Il PROLOG utilizza un sottoinsieme della logica nota come il calcolo dei predicati del primo ordine	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
133	Nell'ambito della programmazione ad oggetti il metodo costruttore non deve:	essere unico	avere lo stesso nome della classe	essere pubblico	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
134	L'Overloading della programmazione ad oggetti consente di:	definire, all'interno della stessa classe, metodi con nome uguale ma parametri distinti	definire, all'interno della stessa classe, metodi con nomi diversi ma parametri uguali	definire, all'interno della stessa classe, metodi con nome diverso, ma stesso comportamento	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
135	La notazione di Backus e Naur (BNF):	Utilizza dei simboli alfanumerici per descrivere gli elementi di una grammatica	Utilizza schemi per classificare le istruzioni di un algoritmo	rappresenta graficamente le strutture dei dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
136	Come funziona Il paradigma di sviluppo conosciuto come TOP DOWN ?	Parte dalla definizione di uno schema generale del progetto, suddividendolo successivamente in elementi più piccoli e di più semplice risoluzione	Si inizia dalla definizione degli elementi atomici e successivamente tramite aggregazione si giunge al progetto completo	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
137	Come funziona Il paradigma di sviluppo conosciuto come BOTTOM UP ?	Si parte dallo sviluppo delle componenti di dettaglio. Queste vengono poi connesse tra loro in modo da formare componenti più grandi, che vengono a loro volta interconnessi fino a realizzare un sistema completo	Parte dalla definizione di uno schema generale del progetto, suddividendolo successivamente in elementi più piccoli e di più semplice risoluzione	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
138	Quale tra i seguenti linguaggi di programmazione utilizza il paradigma della programmazione funzionale?	Lisp	C++	Prolog	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
139	Cosa si intende per progettazione di un test per un software?	E' l'attività di definizione di un insieme di casi di test da sottoporre al Software da testare	E' l'attività di definizione di un insieme di casi di test da sottoporre al personale che utilizzerà il software	E' l'attività di definizione di procedure di sicurezza che dovranno essere seguite nelle attività di utilizzo quotidiano del software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
140	Come avviene la progettazione di un test per un software?	Vengono specificati i dati di input per ogni test, i risultati attesi, ed una descrizione dell'ambiente di esecuzione	Vengono generati dei dati casuali in ingresso al software osservando se avvengono anomalie di funzionamento	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
141	Cosa si intende per "Errore di formato" in un test software?	i parametri di invocazione o di ritorno di una funzionalità sono sbagliati per numero o per tipo	Il software non riesce ad accedere al file system	i parametri di invocazione o di ritorno di una funzionalità sono sbagliati per valore	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
142	Cosa si intende per "Errore di contenuto"? in un test software?	i parametri di invocazione o di ritorno di una funzionalità sono sbagliati per valore	i parametri di invocazione o di ritorno di una funzionalità sono sbagliati per numero o per tipo	Il software non riesce ad accedere al DBMS associato	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
143	Cosa si intende per "Oracolo" in un test software?	Si identifica un metodo, spesso applicato da un agente automatico, per generare i risultati corretti da usare come pietra di paragone durante il test	Si identifica l'operatore che predispone le batterie di test da utilizzare per valutare un software	Un errore non specificato e non associabile a nessuna attività svolta dal software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
144	Cosa si intende per "difetto" di un software?	Corrisponde ad un imperfezione o una mancanza in un sistema software che può dar luogo a un malfunzionamento	Corrisponde alla cattiva documentazione di una funzionalità	Corrisponde al mancato avvio del software sul sistema operativo in uso	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
145	Quando ogni elemento contenuto da un array è a sua volta un array, lo stesso si dice:	Multidimensionale	Unidimensionale	Nulla	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
146	Cosa è una "struttura dati"?	Una struttura dati è un insieme di tipi diversi di dati raggruppati in un'unica dichiarazione	E' un tipo di file memorizzato su file system		PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

147	E' possibile annidare una struttura dati in un'altra?	Si	No	Si, ma solo se si sviluppa in ambiente Linux	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
148	Nei linguaggi di programmazione E' possibile annidare delle istruzioni condizionali per realizzare strutture a scelta multipla?	Si	No	Si, ma solo se si sviluppa in ambiente Linux	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
149	Un istruzione di scelta multipla:	Consente di scegliere fra molte istruzioni in base al valore di un'espressione di selezione	Consente di scegliere al massimo tra due possibili istruzioni in base al valore di un'espressione di selezione	Consente di scegliere tra un numero infinito di istruzioni in base al valore di un'espressione di selezione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
150	In un istruzione di scelta multipla se nessuna etichetta corrisponde al valore dell'espressione:	Viene eseguita l'istruzione di "default" se presente	Viene generato un errore di inconsistenza ed il programma si ferma	Viene generato un errore di Overriding ed il programma si ferma	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
151	E' possibile forzare la fine di un blocco di istruzioni iterate prima che si verifichi la condizione che ne determina la chiusura	Si, se il linguaggio di programmazione prevede l'apposita istruzione	No, in nessun caso	Per chiudere il ciclo è necessario aprire il gestore dei processi sul proprio pc e terminare il processo che esegue il codice	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
152	In informatica ed in particolare quando si parla di sviluppo software la sigla R.A.D. sta per:	Rapid Application Development	Random Access Disk	Redundant Array Disk	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
153	Cosa si intende per programmazione visuale?	E' una tecnica di programmazione che consente di creare un prototipo rapido sviluppando una interfaccia utente a partire da componenti standard (finestre, campi, pulsanti, menu) e associando ad essi delle elaborazioni svolte mediante script	E' un tipo di programmazione in cui si utilizza un editor di testo standard e si scrive il codice che sarà poi passato al compilatore per la verifica di funzionamento	E' una tecnica di programmazione che prevede l'utilizzo di un DBMS associato all'applicazione che si sta sviluppando	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
154	Perché si preferisce utilizzare un ambiente RAD per lo sviluppo di software?	Perché un ambiente RAD offre funzionalità integrate che consente di abbattere anche notevolmente i tempi di sviluppo	Per avere un ambiente di sviluppo più confortevole	Solo utilizzando un ambiente RAD è possibile sviluppare applicazioni web based	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
155	Un ambiente RAD consente un rapido sviluppo di applicazioni tramite:	Una vasta libreria di componenti software già pronti che consentono di creare velocemente un interfaccia grafica, un interfaccia con un database...	Un ampio elenco di pezzi di codice presenti nella documentazione dell'ambiente di lavoro che lo sviluppatore può copiare e incollare	il contatto diretto con un team di sviluppatori direttamente nella chat dell'Ambiente RAD	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
156	In un ambiente RAD possono coesistere applicazioni che utilizzano diversi linguaggi di programmazione?	Si	No	Si, ma solo se si sviluppa in ambiente Windows	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
157	Cosa sono le API?	Sono dei set di definizioni e protocolli con i quali vengono realizzati e integrati software applicativi	Sono dei bug che affliggono un determinato programma, si distinguono in Hardware e Software	Sono quei pezzi di codice definiti "obsoleti" che vanno cioè rimossi quando si aggiorna un software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
158	Quali tra i seguenti è un noto ambiente di sviluppo "Rad"?	Eclipse	Adobe Indesign	Microsoft Project	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
159	In informatica ed in particolare quando si parla di sviluppo software cosa significa la sigla API?	Application Programming Interface	Application Performance Increased	Aided Programming Interface	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
160	Le API sono gratuite o a pagamento?	Alcune sono gratuite altre sono a pagamento	Sono sempre gratuite e ad utilizzo illimitato	Sono sempre a pagamento	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
161	Cosa sono le windows API?	Le Windows API sono l'insieme delle interfacce di programmazione disponibili nei sistemi operativi Windows di Microsoft	Sono l'insieme delle interfacce di programmazione disponibili nei sistemi operativi Linux	Sono le interfacce della Suite Microsoft Office	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
162	In quale contesto si possono utilizzare le API?	Le API sono disponibili in una pluralità di ambienti, sistemi operativi, Servizi on line, linguaggi di programmazione...etc	Solo nelle Web Application	Solo nell'erogazione di servizi OnLine	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

163	Generalmente come si utilizzano le API?	Ogni API espone un interfaccia che consente allo sviluppatore di inviare dei dati e ricevere i risultati	Ogni API riceve i parametri da calcolare tramite email inviata ad un indirizzo dedicato per quell'API	Tramite una pagina dedicata sul sito web del produttore delle API	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
164	Quale tra i seguenti è un esempio pratico di implementazione di una WEB API?	L'app del meteo sul proprio Smartphone	L'app della calcolatrice sul proprio Tablet	L'applicativo Excel sul proprio computer	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
165	In un contesto di programmazione web, cosa indica l'Acronimo "REST"?	Representational State Transfer	Redondant State Transfer	Relational State Transfer	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
166	In un contesto di programmazione web, Cosa è il "REST"?	E' un approccio architetturale alla creazione di WEB API che utilizza il protocollo HTTP	E' un modello di sviluppo di applicazioni web che si basa sulla prototipazione	E' un servizio web di Microsoft che fornisce tutte le API del sistema operativo Windows	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
167	Con che tipo di licenza viene rilasciato Subversion(SVN)?	Licenza open source sotto <u>licenza Apache</u>	Con una licenza commerciale a pagamento	Con una licenza pay-per-use	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
168	La programmazione detta "Imperativa" vede i programmi come:	Sequenze di comandi che agiscono sui dati o sull'ordine di esecuzione delle istruzioni.	Degli oggetti che incorporano sia i dati che i metodi che li gestiscono	Delle WEB API	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
169	La "manutenzione correttiva" durante il ciclo di vita classico di un software viene effettuata:	Durante la fase di gestione dell'applicativo da parte del committente e serve a risolvere eventuali bug riscontrati in produzione	Durante la fase di gestione dell'applicativo da parte del committente e serve ad introdurre nuove funzionalità non previste nella fase di analisi	Prima del rilascio della versione di produzione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
170	Cosa si intende per "Termine del ciclo di vita del software"	La dismissione del software, il backup dei dati e la migrazione verso un nuovo applicativo	Il software viene disattivato in modo che gli utenti non possano accedervi i dati rimangono in memoria	La dismissione del software, l'eliminazione di tutte le istanze installate e di tutti i dati prodotto o gestiti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
171	In quale decennio nacque la programmazione logica?	Negli anni 70	Negli anni 80	Negli anni 90	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
172	Secondo le regole della programmazione logica, come viene descritto un problema?	Viene descritto con un insieme di formule della logica, dunque in forma dichiarativa	richiede, che siano specificate delle rigorose sequenze di passi che, a partire dai dati a disposizione, portino ad ottenere i risultati desiderati	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
173	Come è fatto un programma secondo la concezione della programmazione logica?	Nella programmazione logica i programmi si scrivono descrivendo la conoscenza relativa al problema, cioè specificando gli oggetti che vi intervengono e le relazioni fra di essi	Nella programmazione logica i programmi si scrivono descrivendo le azioni che bisogna compiere per giungere alla risoluzione di un problema e descrivendo i dati che si utilizzano	Nella programmazione logica i programmi sono astrazioni di oggetti. Lo sviluppatore deve descrivere la realtà che implementa come un oggetto con i suoi dati ed i suoi algoritmi	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
174	Le componenti fondamentali di un programma secondo i criteri della programmazione logica sono:	dichiarazioni di fatti sugli oggetti e le loro relazioni, dichiarazioni di regole sugli oggetti e le loro relazioni, domande sugli oggetti e le loro relazioni.	Oggetti, metodi ed attributi	Algoritmi e dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
175	Cosa sono le Subroutine?	Sono delle astrazioni messe a disposizione dai linguaggi di alto livello, e consentono di scrivere porzioni di codice indipendenti, dal programma principale e accessibili con il nome della Subroutine	Sono delle librerie messe a disposizione da alcuni servizi on line che i programmatori possono utilizzare nel proprio codice	Sono delle istruzioni dei linguaggi di alto livello che consentono di interagire con il sistema operativo ed il file system	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
176	Una Subroutine può accettare dei parametri?	E' possibile passare dei parametri alla subroutine e ricevere dei valori di ritorno	No, in nessun caso	No, in nessun caso, ma è possibile ricevere dei valori di ritorno	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

177	In programmazione cosa sono i commenti?	I commenti sono parti del programma che non devono essere interpretate come istruzioni da eseguire, ma servono come indicazioni per una più agevole lettura del programma stesso da parte di chi non ne conosce la logica o anche come promemoria per il programmatore stesso.	Sono parti del programma che contengono dati che devono essere utilizzati dalle istruzioni adiacenti e la cui logica di funzionamento varia a seconda del linguaggio di programmazione utilizzato	Sono parti del programma in cui vengono memorizzate delle procedure o funzioni che sono richiamate nel corpo del programma	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
178	In programmazione cosa si intende per "indentazione del codice"?	Si indica il rientro verso destra delle righe di codice, in maniera tale che già dal primo impatto la struttura logica del programma possa risultare abbastanza chiara	Si indica l'utilizzo corretto delle lettere maiuscole e minuscole per aggiungere maggiore chiarezza al codice	E' quella proprietà che il codice sia aderente allo standard ANSI	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
179	Quale linguaggio di programmazione di alto livello, utilizza l'indentazione del codice per delimitare i blocchi di codice?	Python	Java	Pascal	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
180	In programmazione a cosa serve l'Interprete?	E' un programma informatico che traduce il codice sorgente in codice oggetto, ovvero traduce le informazioni scritte in un linguaggio di programmazione nel linguaggio macchina del computer, le istruzioni sono tradotte una alla volta fino alla fine del file sorgente	E' un linguaggio di programmazione Open Source che funziona secondo le regole della programmazione logica	E' un ambiente di sviluppo di tipo RAD che offre allo sviluppatore una serie di funzionalità grafiche per velocizzare lo sviluppo delle applicazioni. Deve essere associato ad un compilatore per eseguire i programmi sorgenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
181	In programmazione a cosa serve Il Compilatore?	E' un programma informatico che traduce il codice sorgente in codice oggetto, ovvero traduce le informazioni scritte in un linguaggio di programmazione nel linguaggio macchina del computer. Il compilatore traduce tutte le istruzioni del file sorgente prima di passarle alla macchina	E' un ambiente di sviluppo di tipo RAD che offre allo sviluppatore una serie di funzionalità grafiche per velocizzare lo sviluppo delle applicazioni. Deve essere associato ad un interprete per eseguire i programmi sorgenti	Nessuna delle altre risposte è esatta	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
182	Quali sono le differenze tra Interpreti e Compilatori?	Gli interpreti traducono le istruzioni del file sorgente una ad una passandole alla macchina per l'elaborazione, i compilatori traducono tutto il file sorgente contemporaneamente prima di inviarlo per l'elaborazione	I Compilatori traducono le istruzioni del file sorgente una ad una passandole alla macchina per l'elaborazione, gli interpreti traducono tutto il file sorgente contemporaneamente prima di inviarlo per l'elaborazione	Gli interpreti sono dedicati sono ai linguaggi di medio/basso livello come l'assembly. I compilatori sono dedicati ai linguaggi di alto livello	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
183	Nei linguaggi di programmazione le variabili "Globali" sono:	Sono delle variabili visibili in tutto il programma, a meno che non siano ridefinite in una subroutine	Sono delle variabili visibili incondizionatamente in tutto il programma	Sono delle variabili locali alle subroutine, non sono visibili al di fuori di esse	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
184	Cosa rappresenta l'interoperabilità di un Software?	La capacità del software di coesistere e cooperare con altri sistemi	La possibilità di un software ben progettato di essere riutilizzato in tutto o in parte in altre applicazioni	La facilità con cui si interviene in un software per apportare modifiche correttive o per l'introduzione di nuove funzionalità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
185	Nella progettazione di un software cosa si intende per "platform independent"?	E' un modello di un sistema software o di un sistema aziendale indipendente dalla specifica piattaforma tecnologica utilizzata per implementarlo	Indica la possibilità che il software venga utilizzato da una moltitudine di utenti, che per l'accesso utilizzano dispositivi hardware e software diversi tra loro	Indica che il software può essere sviluppato da un team di sviluppatori che utilizza diversi linguaggi di programmazione e/o ambienti di sviluppo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

186	Nello sviluppo di applicazioni Web cosa si intende per Framework?	Un framework è uno strumento di sviluppo web che viene definito come un'applicazione o un insieme di moduli per consentire lo sviluppo agile di applicazioni mediante l'utilizzo di library e/o funzionalità già rilasciate	Un framework è un ambiente di sviluppo completo costituito da un sistema operativo orientato allo sviluppo e da una serie di componenti e librerie armonizzati per supportare lo sviluppatore	Il Framework è un linguaggio di programmazione ad alto livello	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
187	Qual è il vantaggio di utilizzare un Framework nello sviluppo di una applicazione?	L'utilizzo dei framework fa sì che il programmatore non debba ogni volta riscrivere tutto da capo e che si concentri, piuttosto, sul problema da risolvere senza dover implementare funzionalità di uso comune e che già qualcun altro ha provveduto a creare.	E' possibile utilizzare un programmatore con uno "Skill" inferiore e quindi con costi più bassi dato che le funzioni principali sono già implementate nel framework e non devono essere riscritte	Normalmente un Framework impiega componenti hardware o software poco esigenti in termini di risorse, e di conseguenza meno costosi.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
188	L'Architettura di Sviluppo MVC implementa i concetti di:	Modello, Vista, Controllo	Mercato, Vendite, Clienti	Memoria, Velocità, Controllo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
189	Si definisce "Controllo di Qualità" di un Software:	L'insieme degli esami, delle revisioni e dei collaudi svolti nel corso del ciclo di vita del software, il cui scopo è di garantire che il prodotto soddisfi i requisiti imposti;	E' la possibilità di poter eseguire una manutenzione correttiva al software anche in momenti successivi al rilascio	Corrispondono agli stati di avanzamento nelle fasi del ciclo di vita del software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
190	Cosa si intende per "Costi dei difetti" in relazione allo sviluppo software?	Sono i maggiori costi di un progetto software che sono derivati dall'emergere di difetti durante le varie fasi di sviluppo e dei maggiori costi di sviluppo necessari per la loro eliminazione	Sono i costi relativi a materiali difettosi, sia hardware che software, che sono stati riscontrati durante un progetto e che hanno richiesto la loro sostituzione	Sono i costi relativi all'acquisto di materiale non informatico e non relativo a risorse umane per l'esecuzione di un progetto di sviluppo software	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
191	Quale voce non rientra in un progetto di 'Software Quality Assurance' (SQA)?	Cifratura	Audit e revisioni	Documentazione e Reporting	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
192	Quale tra le seguenti NON è un'area di cui si deve occupare il modello del progetto?	La portata-ambito del progetto (scope)	L' architettura	I dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
193	La complessità asintotica del migliore algoritmo di ordinamento di una lista è....	polinomiale.	esponenziale.	lineare.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
194	Si dice che un modulo, che in un software esegue un singolo compito che richiede poca interazione con attività svolte in altre parti del programma, ha...	un'elevata coesione	un'elevata astrazione	un'elevata complessità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
195	Nell'ingegneria del software, il diagramma di relazione tra le entità (entity-relationship diagram)...	Descrive le relazioni tra le strutture dati	Descrive le funzioni che trasformano il flusso dei dati	Descrive come i dati vengono elaborati/trasformati dal sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
196	Nel modello di sviluppo del software a prototipazione incrementale (evolutionary prototyping) quando viene effettuata la ridefinizione dei requisiti nel modello?	Subito prima di costruire un nuovo prototipo	Alla fine del ciclo di vita dell'applicazione	Mai	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
197	Quale non è l'attività non funzionale?	Coordinazione	Usabilità	Accessibilità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
198	Quali delle seguenti sono caratteristiche funzionali di un software?	Conformità e accuratezza	Manutenibilità e portabilità	Efficienza e usabilità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
199	La complessità ciclomatica:	misura direttamente il numero di cammini linearmente indipendenti attraverso il grafo di controllo di flusso	misura la grandezza di un software	misura la difficoltà di scrittura di un algoritmo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

200	Qual è lo scopo di un test di regressione?	Testare le nuove funzionalità e garantire che le funzionalità preesistenti abbiano mantenuto le loro caratteristiche qualitative dopo l'introduzione di queste ultime	Testare le funzionalità dei singoli moduli prima dell'integrazione	Valutare il deterioramento del software all'aumentare dei dati processati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
201	Cosa indica 'il tempo medio fra i guasti' (in inglese mean time between failures, spesso abbreviato in MTBF)?	La misura dell'affidabilità di un sistema	Il tempo necessario a ripristinare un sistema dopo un evento anomalo	Il tempo impiegato da un tester a rilevare un errore nel codice sorgente	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
202	Come si definisce la sintassi di JavaScript che chiama una funzione/metodo?	Espressione di invocazione	Espressione funzionale	Espressione primaria	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
203	L'ordine di grandezza del tempo di esecuzione di un algoritmo fornisce una misura:	dell'efficienza dell'algoritmo stesso	della dimensione dell'algoritmo stesso	della dietrologia di dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
204	Il modello che permette di caratterizzare l'efficienza di esecuzione di un algoritmo è definito come:	complessità computazionale	computazione efficiente	efficacia computazionale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
205	Il modello che permette di caratterizzare l'efficienza di un algoritmo in termini di memoria impiegata è detto:	complessità spaziale	efficienza spaziale	efficacia spaziale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
206	Cosa si intende per complessità computazionale ?	Il modello che permette di caratterizzare l'efficienza di esecuzione di un algoritmo	Il modello che permette di caratterizzare l'efficienza di un algoritmo in termini di memoria impiegata	Tutte le altre risposte sono esatte	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
207	Cosa si intende per complessità spaziale ?	Il modello che permette di caratterizzare l'efficienza di un algoritmo in termini di memoria impiegata	Il modello che permette di caratterizzare l'efficienza di esecuzione di un algoritmo	Tutte le altre risposte sono esatte	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
208	Cos'è la regressione?	E' la possibilità che un programma, in seguito a una modifica dovuta a un intervento di manutenzione o a un processo di sviluppo incrementale, peggiori le proprie funzionalità	Rappresenta la strategia per integrazione dei <i>moduli</i> di un <i>sistema software</i> che fonde le strategie <i>top-down</i> e <i>bottom-up</i>	un sottosistema di un <i>sistema software</i>	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
209	Un algoritmo ricorsivo è trasformabile in modo efficiente in uno iterativo adottando come struttura dati	Uno stack	una coda	un grafo orientato	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
210	In un linguaggio orientato agli oggetti Java-like, si consideri la classe Veicolo e la sua sottoclasse Automobile. Dati gli oggetti <i>V</i> di classe Veicolo e <i>A</i> di classe Automobile, allora l'assegnazione $V = A$:	è sempre corretta	è sempre scorretta	è corretta solo se il valore di <i>A</i> appartiene alla classe veicolo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
211	Cosa si intende per WEB API?	Un insieme di protocolli che consentono la comunicazione di più webapp tra di loro o che consente ad un client esterno di comunicare con una webapp	L'elenco dei parametri che occorre fornire ad una API per ottenere un risultato	Si tratta di una applicazione che funziona unicamente in un browser web	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
212	Secondo il protocollo REST in quali formati avviene lo scambio di informazioni con le WEB API?	Tramite formato JSON o XML	Tramite formato HTML	Tramite formato CSV	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
213	A cosa serve il software Subversion (SVN)?	E' un software di versionamento e controllo di versione prodotto da Apache Software Foundation.Gli sviluppatori lo usano per mantenere le versioni di un file, come ad esempio un codice sorgente, nella versione corrente e in tutta la storia del file stesso.	Si tratta di un ambiente di sviluppo RAD	E' un software dotato di interfaccia grafica che consente allo sviluppatore di implementare un protocollo di comunicazione con una WEB API e verificarne il funzionamento	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

214	Che tipo di dati può ospitare e rendere disponibili ad uno sviluppatore un server Subversion (SVN)?	File sorgenti es. PHP e HTML, ma anche normale documentazione, guide, articoli, e perfino immagini, brani musicali...etc	Solo documentazione in formato PDF, RTF o TXT	Solo elementi multimediali costituiti da Video, immagini..etc	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
215	In informatica, si definisce "PILA" :	E' una struttura dati In cui è possibile inserire degli elementi che andranno tra di loro sovrapposti. Ne consegue che è possibile estrarre gli elementi in ordine inverso rispetto all'ordine in cui sono stati inseriti	E' una struttura nella quale i dati possono essere prelevati nello stesso ordine di quello nel quale sono stati memorizzati, cioè la prima informazione memorizzata è quella che sarà possibile prelevare per prima	E' una struttura in cui i dati vengono inseriti sempre nello stesso ordine ma posso essere prelevata da un punto qualsiasi della pila.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
216	In informatica, si definisce "CODA" :	E' una struttura nella quale i dati possono essere prelevati nello stesso ordine di quello nel quale sono stati memorizzati, cioè la prima informazione memorizzata è quella che sarà possibile prelevare per prima	E' una struttura dati In cui è possibile inserire degli elementi che andranno tra di loro sovrapposti. Ne consegue che è possibile estrarre gli elementi in ordine inverso rispetto all'ordine in cui sono stati inseriti	E' una struttura in cui i dati vengono inseriti sempre nello stesso ordine ma posso essere prelevata da un punto qualsiasi della pila.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
217	Le PILE sono definite con l'acronimo:	LIFO (Last In Firt Out)	FIFO (Firt In First Out)	FINFO (First In Not Firt Out)	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
218	Le CODE sono definite con l'acronimo:	FIFO (Firt In First Out)	LIFO (Last In Firt Out)	FINFO (First In Not Firt Out)	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
219	Quale tra i seguenti è un linguaggio interpretato?	Python	C++	Cobol	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
220	Quale tra i seguenti è un linguaggio Compilato?	C++	Python	Perl	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
221	Cosa si intende per "Porting" di un Software?	E' quel processo di trasposizione, a volte anche con modifiche, di un software, allo scopo di consentirne l'uso su una piattaforma informatica diversa da quella originale	E' l'aumento delle risorse messe a disposizione di un applicativo informatico sia in termini di hardware che software per far fronte a livelli di utilizzo sottostimati durante la fase di analisi	E' quell'attività che si effettua dopo un certo periodo di attività del software volta a correggere eventuali bug rilevati e/o inserire nuove funzionalità richieste dagli utenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
222	Secondo il modello architetturale MVC cosa si intende per "modello"	Il modello rappresenta lo stato della nostra applicazione web (data-base, file, informazioni di sessione, etc.)	Rappresenta i dati di output prodotti da una query	Sono le viste in formato HTML	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
223	Secondo il modello architetturale MVC cosa si intende per "vista"	riproduce anche il contenuto dei modelli che compongono una pagina web, nonché i dati di output	E' il sottosistema che si occupa della gestione dei DBMS	Il modello rappresenta lo stato della nostra applicazione web (data-base, file, informazioni di sessione, etc.)	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
224	Secondo il modello architetturale MVC cosa si intende per "controllo"	controlla le richieste fatte dall'utente, comunica con il Modello (database) e ci mostra la Vista (HTML)	Il modello rappresenta lo stato della nostra applicazione web (data-base, file, informazioni di sessione, etc.)	E' il sottosistema che si occupa dell'accesso degli utenti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
225	In una architettura MVC quale parte del modello si occupa della presentazione dei dati agli utenti?	La componente "Vista"	La componente "Modello"	Le componenti "Modello" e "Vista"	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
226	In una architettura MVC quale parte del modello si occupa della gestione del DBMS?	La componente "Modello"	La componente "Controllo"	Il DBMS viene gestito contemporaneamente da tutte le parti del modello	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
227	Quale dei seguenti aspetti deve essere verificato a livello di test del modulo (unit testing)?	Il funzionamento corretto dell'algoritmo (inclusi i tempi di esecuzione se specificati nei requisiti)	La gestione degli errori	Il percorso di esecuzione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

228	Qual è il maggiore vantaggio del testing d'integrazione bottom-up?	Non è necessario scrivere gli stub	I maggiori punti di decisione son testati all'inizio	Non è necessario scrivere i driver	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
229	Quale delle seguenti NON è una caratteristica che aiuta il testing del software?	Coesione	Osservabilità	Semplicità	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
230	Indicare come si chiama la tecnica di testing che sviluppa situazioni di test (test cases) che hanno lo scopo di verificare/dimostrare che ciascuna funzione del programma è eseguita correttamente.	Black-box testing	Glass-box testing	Grey-box testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
231	La tecnica di testing che sviluppa situazioni di test (test cases) che hanno lo scopo di "esercitare" la logica interna di ciascun modulo software è detta:	white-box testing	black-box testing	grey-box testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
232	Ci sono errori che sfuggono al black-box testing e possono essere individuati con il testing white-box?	Sì, gli errori topografici	Sì, gli errori di comportamento	Sì, gli errori di prestazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
233	Quali sono gli errori che sfuggono al black-box testing, ma possono essere individuati con il testing white-box?	Gli errori logici	Gli errori di comportamento	Gli errori di prestazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
234	Quale delle seguenti tipologie di errori non viene identificato dal testing black-box?	Errori logici	Funzionalità mancanti	Errori di prestazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
235	Il testing del flusso dei dati (Data flow testing) è una tecnica di testing delle strutture di controllo dove il criterio utilizzato per progettare le strutture di test (test cases) è quello di:	selezionare i percorsi di test in base alla posizione e all'uso delle variabili	esercitare tutte le condizioni logiche in un modulo di programma	focalizzarsi sulla validità della struttura dei loop	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
236	Il testing Loop (Loop testing) è una tecnica di testing delle strutture di controllo dove il criterio utilizzato per progettare le strutture di test (test cases) è quello di:	focalizzarsi sulla validità della struttura dei loop	selezionare i percorsi di test in base alla posizione e all'uso delle variabili	esercitare tutte le condizioni logiche in un modulo di programma	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
237	Nello sviluppo di applicazioni Web cosa si intende per "Container"?	Un container è una forma di server virtualizzato in cui non si virtualizza tutta la macchina fisica ma solo l'ambiente di esecuzione delle applicazioni	Un container è un server virtuale che non dispone di sistema operativo il quale deve essere caricato dall'utente a seconda delle sue esigenze	Un container è un server Linux virtualizzato che comprende di serie un Server Web, un ambiente di sviluppo Rad ed un Framework MCV già pronti all'uso.	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
238	Qual è il vantaggio di utilizzare un Container per il deploy di una web app?	Non dovendo inglobare tutte le risorse di un server, in particolare il kernel del sistema operativo, i container sono molto più "leggeri" delle macchine virtuali, richiedono poche risorse di CPU e possono essere attivati in pochi istanti	Avendo all'interno tutte la caratteristiche di una macchina virtuale i container sono degli oggetti completi che non richiedono quasi mai aggiunte e quindi sono utilizzabili senza mai dover fermare il server per aggiornarne le caratteristiche	Possono montare anche un sistema operativo dell'utente accanto a quello dell'Host e sfruttare le capacità di dual boot	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
239	Nello sviluppo di una applicazione, cosa si intende per "information hiding"?	Sono tutti quegli accorgimenti per Impedire l'accesso a dettagli implementativi.	Sono tutti quegli accorgimenti per bloccare l'accesso ad utenti indesiderati	E' un meccanismo dei linguaggi di programmazione che consente ad un oggetto di ereditare le caratteristiche di oggetti posti a livelli superiori	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
240	A cosa serve il modello COCOMO nella progettazione di un Software?	E' un modello matematico utilizzato per stimare alcuni parametri fondamentali come il tempo di consegna e i mesi-uomo necessari per lo sviluppo di un prodotto software	E' un modello visuale che viene condiviso con il committente e consente di verificare congiuntamente lo stato di avanzamento dei lavori	E' un modello di riepilogo di un progetto software che descrive graficamente le risorse necessarie dividendole in risorse già disponibili e risorse da acquisire	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

241	Per analizzare un progetto secondo il modello COCOMO si assume che lo stesso sia sviluppato secondo il modello:	A cascata	Per prototipi	a tecnica libera	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
242	Nella progettazione di un software, l'utilizzo di una "Matrice dei requisiti" :	E' utilizzata per verificare se i requisiti di progetto sono stati raggiunti	E' utilizzata per assegnare le risorse uname ai vari moduli del progetto	Riporta i costi di ogni parte del progetto	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
243	Secondo le linee guida AgiD per lo sviluppo di software per la PA, I dati di accesso (username/password/nome db/ecc..) ai database o a sistemi di altra natura:	Non devono mai essere inseriti nel codice sorgente	Possono essere inseriti nel codice sorgente	Possono essere inseriti nel codice sorgente solo quando si sviluppano applicazioni di tipo Client/Server	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
244	Secondo le linee guida AgiD per lo sviluppo di software per la PA, l'applicazione in esecuzione che privilegi utenti deve utilizzare?	L'applicazione, in esecuzione, non deve utilizzare privilegi amministrativi	L'applicazione, in esecuzione, può utilizzare privilegi amministrativi	L'applicazione, in esecuzione, può utilizzare privilegi amministrativi, solo se opera nella rete LAN della PA	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
245	Secondo le linee guida AgiD per lo sviluppo di software per la PA, cosa si intende per "Input Validation"?	L'applicazione deve assicurare, attraverso opportuni meccanismi di convalida, che tutti i parametri in input, specificati dall'utente, siano congruenti a quanto atteso	L'applicazione deve assicurare la possibilità che vengano bloccati gli input da determinati utenti o classi di utenti	E' la capacità dell'applicazione di continuare a funzionare anche in caso di dati mancanti o incompleti	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
246	Secondo le linee guida AgiD per lo sviluppo di software per la PA, l'accettazione di "caratteri speciali" nella form di input di un software:	Sono considerati pericolosi perché possono innescare delle vulnerabilità	E' lasciato alla scelta degli sviluppatori	Ogni PA, caso per caso, sceglie congiuntamente con gli sviluppatori quali caratteri speciali ammettere per l'input dei dati	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
247	Quale delle tecniche elencate non è una tecnica riconosciuta per il testing del Software?	Destructive testing	Data-flow testing	Domain testing	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
248	La maggior parte degli errori nel software individuati dagli utenti sono dovuti a:	difficoltà nella comprensione del problema	errori di codifica	errori nell'integrazione del sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
249	Cosa si intende per clean coding?	Si tratta di tutta una serie di pratiche e di accorgimenti, non necessariamente legati ad uno specifico linguaggio, il cui obiettivo è rendere il codice sorgente più chiaro da comprendere e con una struttura che renda più agevole apportare modifiche successive	Si tratta di tutta una serie di pratiche e di accorgimenti, non necessariamente legati ad uno specifico linguaggio, il cui obiettivo è rendere il codice sorgente meno chiaro da comprendere	Si tratta di tutta una serie di pratiche e di accorgimenti il cui obiettivo non è quello di creare una struttura che renda più agevole apportare modifiche successive	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
250	Cos'è il refactoring (o code refactoring)?	E' una tecnica strutturata per modificare la struttura interna di porzioni di codice senza modificarne il comportamento esterno	E' una tecnica strutturata per modificare la struttura interna di porzioni di codice che ne modificano anche il comportamento esterno	E' una tecnica strutturata per modificare la struttura esterna di porzioni di codice senza modificarne il comportamento interno	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
251	Come può essere definito un paradigma di programmazione?	E' un modello che permette di descrivere astrattamente l'algoritmo (cioè il metodo di soluzione di un problema)	Non è un modello che permette di descrivere astrattamente l'algoritmo (cioè il metodo di soluzione di un problema)	E' un concetto che non ha alcun tipo di collegamento con i linguaggi di programmazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
252	Quale tra le seguenti rappresenta una corretta definizione di programmazione procedurale?	Paradigma di programmazione che consiste nel creare dei blocchi di codice detti procedure o sottoprogrammi, identificati da un apposito nome, che vengono poi utilizzati dal programmatore nel momento in cui crea programmi più complessi che necessitano di quelle funzioni già preparate	Programmazione in cui il programma è inteso come insieme di istruzioni, ciascuna delle quali può essere pensata come un ordine che viene impartito alla macchina virtuale del linguaggio di programmazione utilizzato	Programmazione utilizzata per rispondere in modo attivo in seguito ad eventi del sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

253	Quale tra le seguenti rappresenta una corretta definizione di programmazione imperativa?	Programmazione in cui il programma è inteso come insieme di istruzioni, ciascuna delle quali può essere pensata come un ordine che viene impartito alla macchina virtuale del linguaggio di programmazione utilizzato	Paradigma di programmazione che consiste nel creare dei blocchi di codice detti procedure o sottoprogrammi, identificati da un apposito nome, che vengono poi utilizzati dal programmatore nel momento in cui crea programmi più complessi che necessitano di quelle funzioni già preparate	Programmazione utilizzata per rispondere in modo attivo in seguito ad eventi del sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
254	Quale tra le seguenti rappresenta una corretta definizione di programmazione ad oggetti?	Paradigma di programmazione che prevede la definizione di oggetti software che interagiscono gli uni con gli altri attraverso scambi di messaggi	Programmazione in cui il programma è inteso come insieme di istruzioni, ciascuna delle quali può essere pensata come un ordine che viene impartito alla macchina virtuale del linguaggio di programmazione utilizzato	Programmazione utilizzata per rispondere in modo attivo in seguito ad eventi del sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
255	Quale tra le seguenti rappresenta una corretta definizione di programmazione funzionale?	Paradigma di programmazione in cui il flusso di esecuzione di un programma assume la forma di una serie di valutazioni di funzioni matematiche	Paradigma di programmazione che prevede la definizione di oggetti software che interagiscono gli uni con gli altri attraverso scambi di messaggi	Programmazione in cui il programma è inteso come insieme di istruzioni, ciascuna delle quali può essere pensata come un ordine che viene impartito alla macchina virtuale del linguaggio di programmazione utilizzato	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
256	Quale tra le seguenti rappresenta una caratteristica della programmazione modulare?	La sua struttura è divisa in moduli, ovvero blocchi di codice che svolgono intere funzioni all'interno del programma	Il flusso di esecuzione di un programma assume la forma di una serie di valutazioni di funzioni matematiche	Le istruzioni impartite sono pensate come un ordine che viene impartito alla macchina virtuale	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
257	Secondo le linee guida AgiD per lo sviluppo di software per la PA, alla dichiarazione di ogni funzione, metodo o classe sviluppato deve sempre precedere un commento che riporti:	Scopo della funzione, Parametri di input e output a/dalla funzione, Valori di ritorno dei parametri di output	Nome del programmatore, Società o AOO di riferimento, email del programmatore	Un set di dati di test da usare come prova, in modo che la funzione venga testata prima dell'utilizzo in produzione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
258	Secondo le linee guida AgiD per lo sviluppo di software per la PA, come deve gestire una applicazione il tracciamento degli eventi?	L'applicazione deve essere predisposta sia per il tracciamento di attività "anomale" sia per le "eccezioni" verificatesi sui sistemi	Non è necessario implementare tecniche di tracciamento sulle applicazioni	Il tracciamento delle applicazioni è una tecnica deprecata in quanto aumenta esponenzialmente la dimensione dei file di log ed in generale appesantisce il sistema	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
259	Secondo le linee guida AgiD per lo sviluppo di software per la PA, la compilazione di una applicazione:	Deve terminare senza errori o warning	Deve terminare senza errori, sono tollerati i Warning	Sono tollerati gli errori solo se non portato a blocchi dell'applicazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
260	Secondo le linee guida AgiD per lo sviluppo di software per la PA, come è necessario gestire gli ambienti di sviluppo, test e produzione?	Gli ambienti di sviluppo, test e produzione devono essere separati fisicamente e/o logicamente.	Gli ambienti di sviluppo e test devono essere separati fisicamente e/o logicamente dall'ambiente di produzione	Possono coesistere se il sistema informatico è sicuro	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
261	Secondo le linee guida AgiD per lo sviluppo di software per la PA, gli strumenti utilizzati in fase di produzione del software possono risiedere su qualche pc che lo utilizza?	Compiler, editor ed altri strumenti di sviluppo non devono essere presenti nei sistemi di produzione in cui l'applicazione risiede	Sì, solo se il sistema informatico è sicuro	Non vi sono indicazioni in merito	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

CONCORSO PUBBLICO A 53 POSTI PER ISPETTORE INFORMATICO DEL CNVVF - PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE

262	Secondo le linee guida AgiD per lo sviluppo di software per la PA, in merito ai sorgenti ed alle librerie utilizzate:	I sorgenti dell'applicazione e delle librerie correlate, fatta eccezione per i linguaggi interpretati, non devono risiedere in testo chiaro all'interno dei sistemi di esercizio, bensì sotto forma di oggetti compilati. Nel caso di linguaggi interpretati, il sorgente dell'applicazione che risiede nei sistemi di esercizio deve essere offuscato.	I sorgenti dell'applicazione e delle librerie correlate, possono risiedere all'interno dei sistemi di esercizio, tuttavia questa attività è sconsigliata perché potrebbe creare confusione	Si lascia libertà in merito al team di sviluppo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
263	Secondo le linee guida AgiD per lo sviluppo di software per la PA, in merito a gestione di password, chiavi e certificati:	Le password mantenute dall'applicazione o le chiavi private dei certificati devono risiedere in container (aree del filesystem, tabelle del database, ecc.) differenti rispetto ai dati dell'applicazione	Le password mantenute dall'applicazione o le chiavi private dei certificati possono risiedere nelle stesse aree in cui sono contenuti i dati dell'applicazione	Le password possono risiedere assieme ai dati dell'applicazione i certificati o le chiavi private devono risiedere su un area diversa	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
264	Secondo le linee guida AgiD per lo sviluppo di software per la PA, lo storage dei dati applicativi:	I dati dell'applicazione memorizzati nel database o nel filesystem devono essere cifrati tramite algoritmi simmetrici con chiave pari almeno a 192 bit (inclusi i bit di parità)	I dati dell'applicazione memorizzati nel database o nel filesystem devono essere cifrati tramite algoritmi simmetrici con chiave pari almeno a 64 bit (inclusi i bit di parità)	I dati dell'applicazione memorizzati nel database o nel filesystem devono essere cifrati tramite algoritmi simmetrici con chiave pari almeno a 128 bit (inclusi i bit di parità)	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
265	Secondo le linee guida AgiD per lo sviluppo di software per la PA, le URL cui accede l'applicazione come devono essere impostati:	Le directory contenenti file o dati di natura personale, critici e sensibili, residenti nella DocumentRoot del web server devono apparire come cifrate nell'URL del client browser	Le directory contenenti file o dati di natura personale, critici e sensibili, residenti nella DocumentRoot del web server possono apparire come cifrate nell'URL del client browser	Le directory contenenti file o dati di natura personale, critici e sensibili, residenti nella DocumentRoot del web server non devono apparire come cifrate nell'URL del client browser	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
266	Secondo le linee guida AgiD per lo sviluppo di software per la PA, in merito al timeout di sessione	L'applicazione deve prevedere il rilascio della sessione utente dopo un certo periodo configurabile di inattività della sessione stessa	L'applicazione deve prevedere il rilascio della sessione utente dopo un tempo prestabilito i 5 minuti di inattività	L'applicazione può non prevedere il rilascio della sessione utente dopo un tempo di inattività	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
267	Secondo le linee guida AgiD per lo sviluppo di software per la PA, al momento del Logout utente:	Quando un utente ha effettuato il log-out, la sessione relativa deve essere invalidata sia sul server che sul client	Quando un utente ha effettuato il log-out, la sessione relativa deve essere invalidata sul client, sul server può essere memorizzata per successivi accessi	Si lascia libertà in merito al team di sviluppo	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
268	Secondo le linee guida AgiD per lo sviluppo di software per la PA, nelle applicazioni web in merito alla scadenza dei Cookie:	Nelle applicazioni web, ciascun cookie generato deve essere soggetto ad un tempo di scadenza oltre il quale non deve più essere considerato valido	Nelle applicazioni web, a seconda degli utilizzi è possibile creare cookie con scadenza e cookie senza limiti di scadenza	Hanno tutti una validità di 30 giorni dalla data di creazione	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
269	Secondo le linee guida AgiD per lo sviluppo di software per la PA, nelle applicazioni web in merito alla gestioni dei Cookie:	i cookie di sessione applicativa devono essere cifrati, non persistent, avere il flag secure attivato e l'attributo HttpOnly impostato	i cookie di sessione applicativa devono essere cifrati non ci sono ulteriori regole	Non vi sono indicazioni in merito	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE
270	Secondo le linee guida AgiD per lo sviluppo di software per la PA, la procedura di accesso ad un sistema informatico:	Non deve con messaggi specifici fornire alcun tipo di aiuto né rendere comprensibile se il processo di autenticazione è fallito a causa del nome utente o della password errata	Deve fornire indicazioni chiare all'utente in caso di autenticazione fallita	Non vi sono indicazioni in merito	PARADIGMI DI SVILUPPO, VERIFICA E RILASCIO SOFTWARE