



CONCORSO PUBBLICO, PER TITOLI ED ESAMI, A N. 1 POSTO DI CATEGORIA D - POSIZIONE ECONOMICA D1 - AREA TECNICA, TECNICO-SCIENTIFICA ED ELABORAZIONE DATI, ANALISTA/SVILUPPATORE JAVA, CON RAPPORTO DI LAVORO SUBORDINATO A TEMPO INDETERMINATO, PRESSO LA DIREZIONE SISTEMI INFORMATIVI D'ATENEО, DA RISERVARE, PRIORITARIAMENTE, ALLE CATEGORIE DI VOLONTARI DELLE FORZE ARMATE IN FERMA BREVE O IN FERMA PREFISSATA DI CUI AL DECRETO LEGISLATIVO N. 66/2010 - BANDITO CON DETERMINA N. 15764/2019 DEL 27/09/2019 E PUBBLICATO SULLA G.U. N. 82 DEL 15/10/2019 - CODICE 20327

## TRACCE PROVE SCRITTE

La Commissione Giudicatrice del concorso, nominata con determina n. 19780/2019 del 15/11/2019, composta da:

PROF. SCARABOTTOLO NELLO - PRESIDENTE

DOTT.SSA ARONA ARIANNA - COMPONENTE

DOTT. PUPILLO VINCENZO - COMPONENTE

SIG.RA ZITO ADDOLORATA - SEGRETARIO

comunica le tracce relative alla prima prova scritta:

### Tema n. 1

L'Ateneo vuole dotarsi di un sistema per la gestione delle sessioni di Laurea.

Una **sessione** di laurea è caratterizzata da una data d'inizio e una data di fine. A essa sono associate una o più **sedute** di laurea. Una **seduta** di laurea è caratterizzata da:

1. data e luogo (ogni luogo è caratterizzato da un codice e da un indirizzo) in cui si svolge (all'interno dell'arco temporale della sessione di laurea);
2. corso di laurea a cui fa riferimento la seduta (un corso di laurea è caratterizzato da un codice e dal nome);
3. insieme degli studenti che partecipano alla seduta (identificati dalla matricola, dal cognome, dal nome, dalla email e dal titolo della tesi di laurea); uno studente può partecipare a una e una sola seduta nel corso di una sessione;
4. insieme dei docenti (identificati dalla matricola, dal cognome, dal nome e dalla email) che svolgono un ruolo nell'ambito della seduta ("presidente", "membro", "aggregato", "supplente").

Ci possono essere più sedute di laurea distinte in una stessa giornata, e più sedute (anche nella stessa giornata) associate a uno stesso corso di laurea (ad esempio può esserci una seduta del corso di laurea di Informatica al mattino e un'altra al pomeriggio).

Ogni corso di laurea viene gestito da una segreteria didattica (caratterizzata da un codice, da una descrizione e da una email) che è responsabile di uno a più corsi di laurea.

Il candidato:

1. rappresenti il modello dei dati del dominio indicato, in Java;



2. descriva una API REST, non autenticata, che consenta di creare un'applicazione web per l'esposizione delle informazioni del modello in oggetto;

descriva una possibile interfaccia web che usando l'API suddetta (con quali chiamate, ecc.) consenta a uno studente di cercare la propria sessione con la relativa seduta, i docenti della commissione e l'email della segreteria a cui chiedere eventuali informazioni.

## Tema n.2

L'Ateneo vuole dotarsi di un sistema per la gestione delle sessioni di Laurea.

Una **sessione** di laurea è caratterizzata da una data d'inizio e una data di fine. A essa sono associate una o più **sedute** di laurea. Una **seduta** di laurea è caratterizzata da:

1. data e luogo (ogni luogo è caratterizzato da un codice e da un indirizzo) in cui si svolge (all'interno dell'arco temporale della sessione di laurea);
2. corso di laurea a cui fa riferimento la seduta (un corso di laurea è caratterizzato da un codice e dal nome);
3. insieme degli studenti che partecipano alla seduta (identificati dalla matricola, dal cognome, dal nome, dalla email e dal titolo della tesi di laurea); uno studente può partecipare a una e una sola seduta nel corso di una sessione;
4. insieme dei docenti (identificati dalla matricola, dal cognome, dal nome e dalla email) che svolgono un ruolo nell'ambito della seduta ("presidente", "membro", "aggregato", "supplente").

Ci possono essere più sedute di laurea distinte in una stessa giornata, e più sedute (anche nella stessa giornata) associate a uno stesso corso di laurea (ad esempio può esserci una seduta del corso di laurea di Informatica al mattino e un'altra al pomeriggio).

Ogni corso di laurea viene gestito da una segreteria didattica (caratterizzata da un codice, da una descrizione e da una email) che è responsabile di uno a più corsi di laurea.

Il candidato:

1. rappresenti il modello dei dati del dominio indicato, in Java ;
2. descriva una API REST, per un backoffice autenticato, che consenta di creare un'applicazione web per l'esposizione delle informazioni del modello in oggetto. L'utente autenticato può vedere solo le informazioni che lo riguardano;
3. descriva una possibile interfaccia web che, usando l'API suddetta (con quali chiamate, ecc.) con una dashboard, consenta di esporre a un docente le informazioni relative alle sedute alle quali dovrà partecipare e le informazioni relative agli studenti coinvolti in tali sedute.

## Tema n. 3

L'Ateneo vuole dotarsi di un sistema per la gestione delle sessioni di Laurea.

Una **sessione** di laurea è caratterizzata da una data d'inizio e una data di fine. A essa sono associate una o più **sedute** di laurea. Una **seduta** di laurea è caratterizzata da:



1. data e luogo (ogni luogo è caratterizzato da un codice e da un indirizzo) in cui si svolge (all'interno dell'arco temporale della sessione di laurea);
2. corso di laurea a cui fa riferimento la seduta (un corso di laurea è caratterizzato da un codice e dal nome);
3. insieme degli studenti che partecipano alla seduta (identificati dalla matricola, dal cognome, dal nome, dalla email e dal titolo della tesi di laurea); uno studente può partecipare a una e una sola seduta nel corso di una sessione;
4. insieme dei docenti (identificati dalla matricola, dal cognome, dal nome e dalla email) che svolgono un ruolo nell'ambito della seduta ("presidente", "membro", "aggregato", "supplente").

Ci possono essere più sedute di laurea distinte in una stessa giornata, e più sedute (anche nella stessa giornata) associate a uno stesso corso di laurea (ad esempio può esserci una seduta del corso di laurea di Informatica al mattino e un'altra al pomeriggio).

Ogni corso di laurea viene gestito da una segreteria didattica (caratterizzata da un codice, da una descrizione e da una email) che è responsabile di uno a più corsi di laurea.

Il candidato:

1. rappresenti il modello dei dati del dominio indicato, in Java ;
2. descriva una API REST per un backoffice autenticato, che consenta di definire una sessione, una seduta di una specifica sessione, e di associare uno studente a una specifica seduta;
3. descriva una possibile interfaccia web che, usando l'API suddetta (con quali chiamate, ecc.) consenta a una segreteria di inserire una sessione, inserire una seduta di una specifica sessione e associarvi uno studente.

La Commissione comunica le tracce relative alla seconda prova scritta a contenuto teorico-pratico:

## Tema 1

### Domanda 1

A partire dal contenuto Json dell'allegato 1, il candidato, usando esclusivamente le classi disponibili nel JDK 8, implementi in Java:

1. l'insieme delle classi che consentono di rappresentare i dati contenuti nel Json in questione;
2. tutti i metodi necessari alla creazione di un oggetto che possa rappresentare il Json in questione, all'aggiunta e all'eliminazione di elementi;
3. a partire dalle classi in questione, i metodi o le classi che consentano di ricostruire un Json analogo a quello presente nell'allegato;
4. un metodo clone() che consenta di ottenere una deep copy degli oggetti.

Il tutto, debitamente documentato e con una corretta gestione degli errori, corredato di un metodo main() con un esempio d'uso della classe stessa.



## Domanda 2

A partire dal contenuto Json dell'allegato 1, il candidato scriva, in Javascript e HTML, una semplice pagina che pubblichi la lista degli insegnamenti che hanno, tra le "formeDidattiche", il "tipo" lezione ("LEZ"). Per ogni insegnamento deve pubblicare il codice, la descrizione (in italiano e/o inglese) con il relativo responsabile (basta il cognome e il nome).

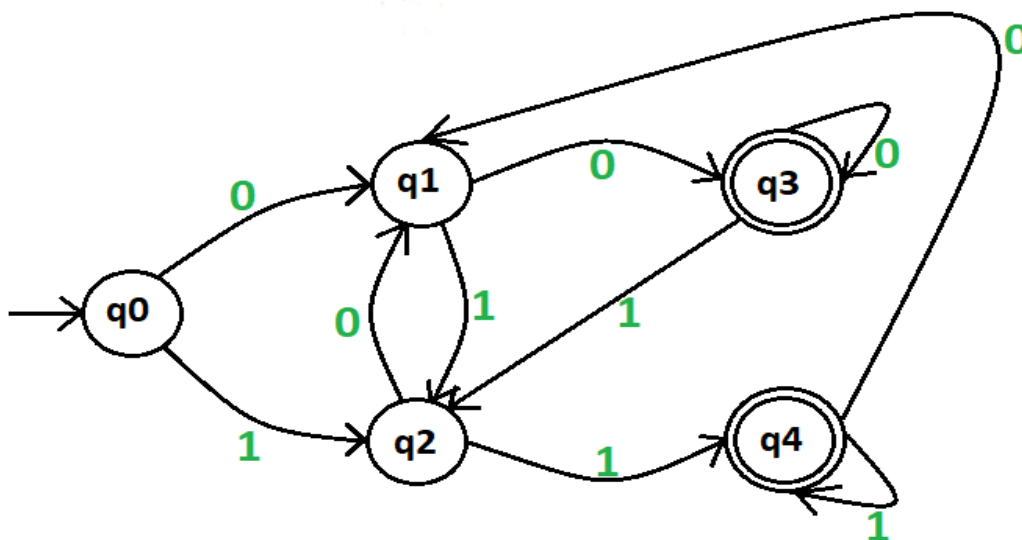
## Domanda 3

Nel 2014 con la pubblicazione del JDK 8, sono state introdotte le lambda-expression. Il candidato fornisca un semplice esempio di uso.

## Tema 2

### Domanda 1

Il candidato implementi in Java usando esclusivamente le classi disponibili nel JDK 8, le classi necessarie a modellare il seguente automa a stati finiti:



La classe/classi devono implementare:

1. i metodi necessari per passare da uno stato al successivo;
2. i metodi necessari per stampare il contenuto dello stato corrente;
3. i metodi necessari per stampare (su stdout) la descrizione (con un formalismo a scelta del candidato) dell'automa stesso;



4. i metodi necessari per consentire all'automa di tornare indietro di N passi rispetto allo stato attuale.

Il tutto, debitamente documentato e con una corretta gestione degli errori, corredato di un metodo main() con un esempio d'uso della classe stessa.

## Domanda 2

A partire dal contenuto Json dell'allegato 1, il candidato scriva, in Javascript e HTML, una semplice pagina che pubblichi la lista degli insegnamenti. Per ogni insegnamento deve pubblicare il codice, la descrizione (in italiano e/o inglese) con i relativi sottoscrittori, cioè i corsi di laurea che prevedono l'erogazione di tale insegnamento. Dei sottoscrittori deve pubblicare il codice, la descrizione.

## Domanda 3

Il candidato descriva lo statement Java try-with-resources comparso con il JDK 7.

## Tema 3

### Domanda 1

Il candidato implementi in Java (JDK 8) una struttura dati di tipo Stack che memorizzi un payload di tipo Object, senza l'utilizzo delle Java Collection Framework. È necessario fornire una propria implementazione.

La classe deve implementare i seguenti metodi:

- push(Object o), inserisce un oggetto nello stack;
- pop(), rimuove un oggetto dallo stack e lo restituisce;
- top(), restituisce l'elemento in cima allo stack, senza rimuoverlo;
- clone(), restituisce un clone dello Stack;
- isEmpty(), restituisce true se lo stack è vuoto;
- depth(), restituisce il numero di elementi presenti nello stack;
- toJson(), restituisce la versione serializzata in Json del contenuto dello stack (con chiavi definibili a piacere).

Il tutto, debitamente documentato e con una corretta gestione degli errori, corredato di un metodo main() con un esempio d'uso della classe stessa.

### Domanda 2

A partire dal contenuto Json dell'allegato 1, il candidato scriva, in Javascript e Html, una semplice pagina che pubblichi la lista degli insegnamenti. Per ogni insegnamento deve pubblicare il codice, la descrizione (in italiano e/o inglese) e le lingue in cui viene erogato l'insegnamento.



## Domanda 3

Il candidato descriva la metodologia di sviluppo Waterfall.

### Allegato 1

```
{
  "payload": [
    {
      "annoAccademico": "2020",
      "codice": "F94-60",
      "cfu": 6,
      "descrizione": "Information management",
      "descrizioneEng": "INFORMATION MANAGEMENT",
      "strutturaResponsabile": {
        "codice": "F94",
        "tipo": "CDS"
      },
      "corsoDiStudio": {
        "codice": "F94",
        "tipo": "CDS"
      },
      "lingue": [
        {
          "lingua": "eng"
        },
        {
          "lingua": "ita"
        }
      ],
      "listaSettori": [
        {
          "codice": "INF/01",
          "descrizione": "Informatica",
          "cfu": 6
        }
      ],
      "obiettivi": "L'insegnamento ha l'obiettivo di introdurre concetti avanzati legati alla gestione...",
      "obiettiviEng": "The objective of this course is to introduce advanced concepts for information ...",
      "oreTotali": 48,
      "periodo": {
        "periodoErogazione": 2,
        "tipoPeriodoErogazione": "semestre",
        "durata": 1,
        "tipoDurata": "semestre"
      },
      "edizioni": [
        {
          "_relatedLinks": {
            "links": [
              {
                "rel": "programma",
                "href":
"https://apps.unimi.it/ws/offertadidattica/programmaDidattico/2020%2FF94/F94-60.20.1"
              }
            ]
          },
          "codice": "F94-60.20.1",
          "responsabile": {
            "nome": "SARA",

```



# UNIVERSITÀ DEGLI STUDI DI MILANO

```
"cognome": "FORESTI",
"email": "SARA.FORESTI@UNIMI.IT"
},
"periodo": {
  "periodoErogazione": 2,
  "tipoPeriodoErogazione": "semestre"
},
],
"sottoscrittori": {
  "offerte": [
    {
      "_relatedLinks": {
        "links": [
          {
            "rel": "offerta",
            "href":
"https://apps.unimi.it/ws/offertadidattica/offertaFormativa/2020/F94of2"
          }
        ]
      },
      "codice": "F94of2",
      "descrizione": "Informatica (Classe LM-18)-immatricolati dall'a.a. 2014/2015",
      "descrizioneEng": "Computer Science (Classe LM-18)-enrolled from 2014/2015 academic
year",
      "sitoCorso": "https://informatica-lm.cdl.unimi.it/"
    },
    {
      "_relatedLinks": {
        "links": [
          {
            "rel": "offerta",
            "href":
"https://apps.unimi.it/ws/offertadidattica/offertaFormativa/2020/F2Yof2"
          }
        ]
      },
      "codice": "F2Yof2",
      "descrizione": "Sicurezza informatica (Classe LM-66)-immatricolati dall'a.a. 18/19",
      "descrizioneEng": "Informatics Security (Classe LM-66)-Enrolled from 18/19 academic
year",
      "sitoCorso": "https://sicurezzainformatica.cdl.unimi.it/"
    }
  ]
},
"formeDidattiche": [
  {
    "descrizione": "Lezioni",
    "tipo": "LEZ",
    "oreDocente": "48",
    "oreIndividuali": "102"
  }
],
],
"messages": null,
"errors": null
}
```



# UNIVERSITÀ DEGLI STUDI DI MILANO

## LA COMMISSIONE

PROF. SCARABOTTOLO NELLO - PRESIDENTE

.....

DOTT.SSA ARONA ARIANNA - COMPONENTE

.....

DOTT. PUPILLO VINCENZO - COMPONENTE

.....

SIG.RA ZITO ADDOLORATA - SEGRETARIO

.....