

Употреба штампане литературе и комуникационих уређаја није дозвољена и строго је кажњива током трајања испита.

ПРВИ ДЕО ИСПИТА (30 минута)

1. Један програмер, желећи да реализује узорак убацивања зависности (*Dependency Injection*), употребио је контејнер са инверзијом контроле на следећи начин да инстанцира класу Корисник, којој је потребна конекција ка бази:

```
class Korisnik {
    protected $db;
    public function __construct() {
        $this->db = IoC::resolve('db_conn');
    }
    /* ostali metodi klase Korisnik */
}
$kor = new Korisnik();
```

а) [3] Дати комплетну исправну декларацију IoC класе као генеричког регистра (навести исправне декларације метода са параметрима, а уместо имплементација тела метода могу се ставити коментари).

б) [3] Објаснити у чему је програмер направио грешку у датом коду. Написати исправну варијанту класе Корисник и показати како се на исправан начин она инстанцира користећи IoC класу.

2. [5] Користећи *SVN (Subversion)* систем, навести редослед свих потребних команди, које је потребно извршити и објаснити ситуацију која доводи до конфликта, а затим написати команде и објаснити како се конфликт може разрешити. Претпоставити да постоје два корисника који користе исти *SVN* репозиторијум, да у репозиторијуму постоје бар два фајла и да на почетку корисници немају радне - локалне копије фајлова из репозиторијума. У сваком тренутку навести и која је верзија фајла (ревизија) са којом корисник ради.

3. [5] За сваку реченицу у наставку, оценити да ли представља добро формулисан захтев или не и зашто:

А.Систем треба да буде једноставан за коришћење.

Б.Систем треба да одговори у мање од 2 секунде исписом резултата.

Ц.Систем би могао да обради датотеке до 2 МБ величине.

Д.Систем ће одговорати свим захтевима стандарда IEEE 854.

Е.Систем треба да ради на свим веб браузерима.

Употреба штампане литературе и комуникационих уређаја
није дозвољена и строго је кажњива током трајања испита.

Име и презиме:	Број индекса:
----------------	---------------

ДРУГИ ДЕО ИСПИТА (150 минута)

4. Дата је следећа спецификација веб система за пријављивање практичне наставе. Корисници система су наставници и студенти. Осим основних информација које поседује корисник (јединствено корисничко име, лозинка, име и презиме), наставник додатно има своје звање (редовни проф, ванредни проф, доцент или асистент) и обавезну припадност одређеној катедри. Свака катедра има свој назив и опционо може имати један или више одсека којима руководи. Студент има број свој индекса, годину студија и одсек коме припада.

Наставник држи предмет(е) у одређеној школској години. Чува се информација која је текућа школска година. Сваки предмет има своју шифру, назив, број кредита (ЕСПБ), семестар коме припада и фонд часова П+В+Л (предавања, вежби и лабораторије). Студент у одређеној школској години прати одређени предмет. За предмете које прати може да се пријави на неки облик практичне наставе и за сваку пријаву треба чувати датум и време пријављивања. Практична настава се везује за одређени предмет и може бити у форми пројекта или лабораторијске вежбе. Пројекат има свој назив, тип (индивидуални или тимски), и опционо максималан број чланова тима (број између 2 и 4). Лабораторијска вежба има свој редни број и термин одржавања, изражен у броју наставне недеље у којој се организује. Сматрати да се лабораторијска вежба ради индивидуално. Наставник за предмет који држи у текућој школској години може да формира неки облик практичне наставе и на тај начин отвори пријаву за ту активност. Студенти су онда у могућности да се пријаве за ту активност (лаб.вежба/пројекат) на таквом предмету, под условом да прате баш тај предмет. Пријава је могућа све док у неком тренутку наставник не затвори пријаву. Уколико се студент пријављује за пројекат који је тимски, додатно треба чувати податке о саставу тима (односно само о другим члановима, које је вођа – капитен тима одабрао). Треба водити рачуна да ће систем избацити грешку, уколико капитен тима покуша да у тим убаци неког студента који је већ пријављен у оквиру неког другог тима или покуша да убаци истог студента више пута. У том случају, капитену омогућити да понови унос, али само тог студента, уз задржавање података о оним члановима који су валидни. Уколико се студент пријављује за лабораторијску вежбу, треба му дозволити да одабере жељени радни дан када хоће да је ради (бира нпр. из падајуће листе: понедељак – петак).

- а) [14] На нивоу релационе базе података, дате у *MySQL etfprijava2019.sql*, утврдити да ли је база података и да ли су подаци у бази у складу са горе описаном спецификацијом. Урадити све измене, како би база података одговарала овим корисничким захтевима. Модификовану базу попунити подацима, неопходним за тачке б) и д).

Излаз задатка 4а): скрипт модификоване базе података *etfprijava2019-GGGGBBBB.sql*

- b) [14] У алату *NetBeans IDE*, реализовати *MVC (Model-View-Controller)* апликацију коришћењем програмског језика *PHP*. Имплементацију је могуће радити коришћењем објектно-оријентисаног *PHP* кода или коришћењем *CodeIgniter* оквира. Потребно је имплементирати:
- део система који ради пријаву за пројекат од стране студента. Уколико је пројекат предвиђен за тимски рад, потребно је одабрати преостале чланове тима и вршити проверу свих дефинисаних ограничења.
 - део система који омогућава да наставник (након успешног логовања) за већ отворене пријаве за практичну наставу на неком од његових предмета, затвори пријаву (пријава се не брише из система, већ само архивира, тако да се студенту на даље онемогућује да се пријави на ту активност).

Излаз задатка 4б): пројекат *PHP* имплементације описаног дела апликације (уколико се ради у радном оквиру, прекопирати такође читав пројекат)

- c) [8] У алату *Selenium IDE* у веб прегледачу *Mozilla Firefox* наснимити скрипт који садржи тестове који тестирају дату форму за логовање и тестирање Ваше реализоване имплементације пријаве за пројекат, под тачком (б). Тестирање треба извршити методом црне кутије. Обухватити главни (успешан) сценарио и све алтернативне гране сценарија логовања и сценарија пријаве за пројекат. Сваки тест мора имати бар једну верификациону тачку. Скрипт треба да има минималан број тестова.

Излаз задатка 4ц): експортирани фајл из *Selenium-a* – *testoviGGGGBBBB.side*

- d) [8] Нацртати дијаграм секвенце за сценарио претраживања и попуњавања пријаве за пројекат или лабораторијску вежбу, након што се студент успешно улоговао. Форма за претраживање садржи текстуално поље за унос назива предмета и 2 чекбокса: лабораторијске вежбе и пројекат. Мора се унети бар један податак за успешну претрагу. Након што потврди форму, студент добија све тренутно активне пријаве за лаб.вежбе / пројекат, које задовољавају параметре претраживања. Излиставају се и оне пријаве за предмете које студент тренутно не слуша, али су отворене. Студент затим одабира једну (на коју се може пријавити) и извршава попуњавање евентуално додатних информација (нпр. ако је пројекат тимски, тражити и додатне информације о члановима тима или исписати одређену поруку са грешком).

Реализовани дијаграм секвенце треба да приказује све серверске компоненте (M,C) и клијентске компоненте (V), као и *HTML* форме за описани сценарио.

Излаз задатка 4д): нацртати реализован дијаграм на додатном папиру (тачке (б) и (ц) не узимати у обзир, већ само опис спецификације задатка и опис сценарија под овом тачком!)

Име и презиме:	Број индекса:
----------------	---------------

Решење задатка 4д: