

**КОНСПЕКТ
ЗА
ДЪРЖАВЕН ИЗПИТ ЗА ЗАВЪРШВАНЕ НА
ОБРАЗОВАТЕЛНО-КВАЛИФИКАЦИОННА
СТЕПЕН „БАКАЛАВЪР”**

**СПЕЦИАЛНОСТ
„СОФТУЕРНО ИНЖЕНЕРСТВО”**

Промените в конспекта за ДИ са приети с
решения на ФС: протокол № 16/29.11.2021 г.,
протокол № 1/24.01.2022 г.
протокол № 2/28.02.2022 г.

СЪДЪРЖАНИЕ

УКАЗАНИЯ ЗА ПРОВЕЖДАНЕТО НА ДЪРЖАВЕН ИЗПИТ	3
КОНСПЕКТ ЗА ДЪРЖАВЕН ИЗПИТ ЗА СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО”	4
АНОТАЦИИ НА ВЪПРОСИТЕ	6
ЛИТЕРАТУРА	16

УКАЗАНИЯ ЗА ПРОВЕЖДАНЕТО НА ДЪРЖАВЕН ИЗПИТ ЗА ЗАВЪРШВАНЕ НА ОБРАЗОВАТЕЛНО-КВАЛИФИКАЦИОННА СТЕПЕН „БАКАЛАВЪР”

СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО”

При явяване на държавен изпит всеки студент е длъжен да носи студентската си книжка, да се яви навреме пред предварително оповестената зала и да спазва указанията на квесторите за настаняване в залата.

Държавният изпит по специалност „*Софтуерно инженерство*” е в две части, които се провеждат в два дни. През първия ден изпитът е практически (решаване на задачи). На втория ден изпитът е теоретичен.

По време на всяка една част от изпита листата за писане са осигурени и подпечатани от ФМИ, други не се внасят. Пише се само с химикал - задължително син или черен цвят. Молив може да се използва само за чертежи.

По време на изпита може да се използва официално издадено копие на конспекта (получава се от квесторите; собствено копие не може да се внася). Всички други пособия са забранени.

Забранено е използването на електронни устройства от всякакъв вид. Необходимо е всички внесени мобилни устройства и компютърна техника да бъдат изключени преди започване на изпита и да бъдат оставени на определените за целта места. Намирането при студентите на нерегламентирани помощни средства се счита за опит за преписване. По време на изпита не се водят разговори, не се пуши и не се излиза от залата.

Работите се оценяват от комисия. Практическият и теоретичният изпит се оценяват поотделно. При положение, че и на двата изпита оценката е по-голяма или равна на 3.00, то крайната оценка от държавния изпит е закръглената по правилата средно аритметична оценка от двата изпита. В противен случай оценката е слаб (2.00). Оценката се закръгля до втори знак след десетичната запетая. Оценките са окончателни и не подлежат на преразглеждане.

Според правилника на СУ студентите нямат право на явяване за повишаване на оценка от ДИ, ако той е бил успешно положен. Напомняме на студентите, че според ЗВО за продължаване на образованието в ОКС „Магистър” (**срещу заплащане**) е необходима оценка най-малко „добър” от дипломата за ОКС „Бакалавър”.

КОНСПЕКТ ЗА ДЪРЖАВЕН ИЗПИТ ЗА СПЕЦИАЛНОСТ „СОФТУЕРНО ИНЖЕНЕРСТВО”

ОСНОВИ И ЯДРО НА КОМПЮТЪРНИТЕ НАУКИ

1. Графи. Дървета. Обхождане на графи.
2. Булеви функции. Пълнота.
3. Крайни автомати. Регулярни езици. Теорема на Клини.
4. Контекстно-свободни граматики и езици. Стекови автомати.
5. Компютърни архитектури. Формати на данните. Вътрешна структура на централен процесор – блокове и конвейерна обработка, инструкции.
6. Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.
7. Процеси и комуникация между тях в операционната система.
8. Компютърни мрежи и протоколи – OSI модел. Протоколи IPv4, IPv6, TCP, HTTP .
9. Бази от данни. Релационен модел на данните.
10. Бази от данни. Нормални форми.
11. Процедурно програмиране – основни конструкции
12. Процедурно програмиране – указатели, масиви и рекурсия
13. Обектно-ориентирано програмиране. Основни принципи. Класове и обекти. Наследяване и капсулация.
14. Обектно-ориентирано програмиране. Подтипов и параметричен полиморфизъм. Множествено наследяване.
15. Структури от данни. Стек, опашка, списък, дърво. Основни операции върху тях. Реализация.

СОФТУЕРНО ИНЖЕНЕРСТВО

16. Софтуерно инженерство и неговото място като дял от знанието. Софтуерен процес и модели на софтуерни процеси. Концепция за многократна употреба.
17. Модели на разпределени софтуерни архитектури. Среда и протоколи за разпределени приложения.
18. Модели и методи за проектиране на потребителски интерфейс.
19. Управление на качеството на софтуерни приложения. Тестване на софтуер
20. Софтуерна архитектура. Проектиране и документиране на софтуерни архитектури.
21. Инженеринг на софтуерните изисквания. Техники за извличане, анализ и валидиране на софтуерните изисквания. Специфициране на изискванията.
22. Проектиране и интегриране на софтуерни системи

- 23. Планиране на проекта – същност и основни елементи, обхват на проекта, времеви и финансови ресурси. Дейности по управление и контрол, методи и средства за създаване на план-график на проекта
- 24. Използване на XML за структуриране, валидация, обработка и представяне на документно съдържание.

МАТЕМАТИКА И ПРИЛОЖЕНИЯ

- 25. Теорема на Ферма. Теорема за средните стойности (Рол, Лагранж и Коши). Формула на Тейлър.
- 26. Определен интеграл. Дефиниция и свойства. Интегрируемост на непрекъснатите функции. Теорема на Нютон-Лайбниц
- 27. Ранг на система вектори. Ранг на матрица. Теорема за равенството на ранг на матрица, ранг на системата от вектор-редове и ранг на системата от вектор-стълбове на тази матрица. Системи линейни уравнения – теорема на Руше, фундаментална система от решения за хомогенна система и теорема за размерността на подпространството от решения на хомогенна система.
- 28. Полиноми на една променлива – определение, действия с полиноми и степен на полином. Теорема за делене на полиноми с частно и остатък, правило на Хорнер. Най-голям общ делител на полиноми – твърдение на Безу и алгоритъм на Евклид. Корени на полином и зависимости между корените и коефициентите на полином (формули на Виет).
- 29. Дискретни разпределения. Равномерно, биномно, геометрично и Пуасоново разпределение. Задачи, в които възникват. Моменти – математическо очакване и дисперсия.

АНОТАЦИИ НА ВЪПРОСИТЕ

1. Графи. Дървета. Обхождане на графи.

Дефиниции за краен ориентиран (мулти)граф и краен неориентиран (мулти)граф. Дефиниции за маршрут (контур) в ориентиран мултиграф и път (цикъл) в неориентиран мултиграф. Свързаност и свързани компоненти на граф. Дефиниция на дърво и кореново дърво. Доказателство, че всяко кореново дърво е дърво и $|V|=|E|+1$. Покриващо дърво на граф. Обхождане на граф в ширина и дълбочина. Ойлерови обхождания на мултиграф. Теорема за съществуване на Ойлеров цикъл (с доказателство) и Ойлеров път.

Литература: [12].

2. Булеви функции. Пълнота.

Дефиниция на булева функция (БФ) и формула над множество БФ. БФ с 1 и 2 променливи. Свойства. Дефиниция на пълно множество БФ. Формулировка и доказателства на теоремата за разбиване на БФ по част от променливите и теоремата на Бул. Теорема на Пост.

Литература: [12].

3. Крайни автомати. Регулярни езици. Теорема на Клини.

Детерминирани крайни автомати. Регулярни операции. Недетерминирани крайни автомати. Представяне на всеки недетерминиран краен автомат с детерминиран (с доказателство). Затвореност относно регулярните операции. Теорема на Клини (с доказателство). Лема за покачването (uv^nw) (с доказателство). Примери за регулярни и нерегулярни езици. Минимизация на състоянията. Теорема на Майхил-Нероуд (с доказателство). Алгоритъм за конструиране на минимален автомат, еквивалентен на даден детерминиран краен автомат.

Литература: [12], [40].

4. Контекстно-свободни граматики и езици. Стекови автомати.

Контекстно-свободни граматики. Дървета за синтактичен анализ. Нормална форма на Чомски. Стекови автомати. Връзка между стековите автомати и контекстно-свободните граматики (доказателство в едната посока по избор). Свойства на затвореност. Лема за покачването (xuv^nw) (с доказателство). Примери за езици, които не са контекстно-свободни.

Литература: [12], [40].

5. Компютърни архитектури. Формати на данните. Вътрешна структура на централен процесор – блокове и конвейерна обработка, инструкции.

Обща структура на компютрите и концептуално изпълнение на инструкциите, запомнена програма.

Формати на данните

- цели двоични числа;
- двоично-десетични числа;
- двоични числа с плаваща запетая;
- символни данни и кодови таблици.

Вътрешна структура на централен процесор

- регистри;
- аритметико-логическо устройство;

- регистър на състоянието и флагове;
- блок за управление.

Инструкции на централен процесор

- префикси;
- код на операцията;
- местоположение на операндите;
- модели на адресация на операндите;
- аритметико-логически инструкции;
- низови инструкции;
- безусловни и условни преходи;
- управление на програмата.

Литература: [1].

6. Структура и йерархия на паметта. Сегментна и странична преадресация. Система за прекъсване – приоритети и обслужване.

Йерархия на паметта - кеш-памет, оперативна памет и виртуална памет.

Сегментна преадресация

- сегментен селектор;
- сегментен дескриптор;
- сегментни таблици и регистри.

Странична преадресация

- каталог на страниците;
- описател на страница;
- стратегии на подмяна на страниците.

Прекъсвания

- структура и обработка;
- типове прекъсвания;
- конкурентност и приоритети;
- обслужване на прекъсванията;
- контролери на прекъсванията.

Литература: [1].

7. Процеси и комуникация между тях в операционната система.

Процеси и комуникационни канали - основни абстракции, предоставени от операционната система. Неформално определение и функционални изисквания.

Комуникация между процеси чрез споделена памет. Съревнование за ресурси (race condition). Хардуерна защита на ресурс, синхронизация чрез spinlock.

Синхронизация от високо ниво – семафор. Приспиване и събуждане на процеси (block/wakeup). Реализация чрез семафори на комуникационна тръба (pipe), съхраняваща n елемента.

Литература: [LBS, стр. 1-80], [MOS, втора глава], [OSC, шеста глава]

Примерни задачи:

Задача 1. Процесът P и няколко копия на процеса Q изпълняват поредица от три инструкции:

process P	process Q
p_1	q_1
p_2	q_2
p_3	q_3

Осигурете чрез семафори синхронизация на P и копията на Q така, че инструкцията p_1 да се изпълни преди инструкцията q_2 на всяко копие на Q. Освен това искаме след изпълнението на p_1, всички копия на Q да завършат изпълнението си.

Задача 2. Всеки от процесите P,Q и R изпълнява поредица от три инструкции:

process P	process Q	process R
p_1	q_1	r_1
p_2	q_2	r_2
p_3	q_3	r_3

Осигурете чрез семафори синхронизация на P, Q и R така, че да се изпълнят следните изисквания:

- (а) Инструкцията p_1 да се изпълни преди q_2 и r_2.
- (б) Ако q_2 се изпълни преди r_2, то и q_3 да се изпълни преди r_2.
- (в) Ако r_2 се изпълни преди q_2, то и r_3 да се изпълни преди q_2.

Задача 3: Всеки от процесите P и Q изпълнява поредица от три инструкции:

process P	process Q
p_1	q_1
p_2	q_2
p_3	q_3

Осигурете чрез семафори синхронизация на P и Q така, че инструкцията p_1 да се изпълни преди q_2, а q_2 да се изпълни преди p_3.

Литература:

[LBS] The little Book of Semaphores

<http://greenteapress.com/semaphores/LittleBookOfSemaphores.pdf>

[MOS]Tanenbaum, A., Bos,H., Modern Operating Systems, 4th ed. Pearson, 2014

[OSC] Silberschatz, Galvin, and Gagne, Operating System Concepts, 7th Edition

8. Компютърни мрежи и протоколи – OSI модел. Протоколи IPv4, IPv6, TCP, HTTP.

OSI модел – най-обща характеристика на нивата, съпоставяне с модела TCP/IP. IPv4 адресация – класова и безкласова. Основни характеристики на протокол IPv6. TCP – процедура на трикратно договаряне. Хипертекстов протокол HTTP.

Литература: [2], [39], [52].

9. Базы от данни. Релационен модел на данните.

Релационен модел на данните: домейн; релация; кортежи; атрибути; схема на релация; схема на релационна база от данни; реализация на релационната база от данни; видове операции върху релационната база от данни; заявки към релационната база от данни. Релационна алгебра: основни (обединение; разлика; декартово произведение; проекция; селекция) и допълнителни (сечение; частно; съединение; естествено съединение) операции.

Примерни задачи: Съставяне на SQL заявки, DDL и DML команди.

Литература: [31].

10. Базы от данни. Нормални форми.

Нормални форми. Проектиране схемите на релационните бази от данни. Аномалии, ограничения, ключове. Функционални зависимости, аксиоми на Армстронг. Първа,

втора, трета нормална форма, нормална форма на Бойс-Код. Многозначни зависимости; аксиоми на функционалните и многозначните зависимости; съединение без загуба; четвърта нормална форма.

Примерни задачи: Привеждане на схема на базата от данни (при зададени функционални зависимости) към зададена нормална форма.

Литература: [31]

11. Процедурно програмиране – основни конструкции

1. Принципи на структурното програмиране.
2. Управление на изчислителния процес. Основни управляващи конструкции – условни оператори, оператори за цикъл.
3. Променливи – видове: локални променливи, глобални променливи; инициализация на променлива; оператор за присвояване.
4. Функции и процедури. Параметри – видове параметри. Предаване на параметри – по име и по стойност. Типове и проверка за съответствие на тип
5. Символни низове. Представяне в паметта. Основни операции със символни низове.

Забележка: По този въпрос е възможно да бъдат дадени задачи за практическата част на изпита.

Литература: [22], [25], [49].

12. Процедурно програмиране – указатели, масиви и рекурсия.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Указатели и указателна аритметика.
2. Едномерни и многомерни масиви. Основни операции с масиви – индексирание. Сортиране и търсене в едномерен масив – основни алгоритми.
3. Рекурсия – пряка и косвена рекурсия, линейна и разклонена рекурсия.

Забележка: По този въпрос е възможно да бъдат дадени задачи за практическата част на изпита.

Литература: [22], [25], [49].

13. Обектно-ориентирано програмиране. Основни принципи. Класове и обекти. Наследяване и капсулация.

1. Абстракция със структури от данни. Класове и обекти. Декларация на клас и декларация на обект. Основни видове конструктори. Управление на динамичната памет и ресурсите (“RAII”). Методи – декларация, предаване на параметри, връщане на резултат.
2. Наследяване. Производни и вложени класове. Достъп до наследените компоненти. Капсулация и скриване на информацията. Статични полета и методи.

Забележка: По този въпрос е възможно да бъдат дадени задачи за практическата част на изпита.

Литература: [21], [22], [24], [49].

14. Обектно-ориентирано програмиране. Подтипов и параметричен полиморфизъм. Множествено наследяване.

1. Виртуални функции и подтипов полиморфизъм. Динамично свързване. Абстрактни методи и класове. Масиви от обекти и от указатели към обекти.
2. Параметричен полиморфизъм. Шабини на функция и на клас
3. Множествено наследяване.

Забележка: По този въпрос е възможно да бъдат дадени задачи за практическата част на изпита.

Литература: [21], [22], [24], [49].

15. Структури от данни. Стек, опашка, списък, дърво. Основни операции върху тях. Реализация

1. Структури от данни – дефиниране на понятието.
2. Списък. Логическо описание. Списък с една и две връзки. Характеристики на реализациите с една и две връзки. Сложност на операциите по добавяне, премахване и намиране на елемент. Дефиниране на клас за списък, използващ една от реализациите.
3. Стек. Логическо описание. Характеристики на статичната, динамичната и свързаната реализация. Сложност на операциите по добавяне и премахване на елемент. Дефиниране на клас за стек, използващ една от реализациите.
4. Опашка. Логическо описание. Характеристики на статичната, динамичната и свързаната реализация. Сложност на операциите по добавяне и премахване на елемент. Дефиниране на клас за опашка, използващ една от реализациите.
5. Дървовидни структури от данни – кореново дърво и двоично кореново дърво. Логическо описание. Начини за представяне в паметта. Дефиниране на клас, реализиращ кореново дърво или двоично кореново дърво.
6. Двоично кореново дърво за търсене. Логическо описание. Начини за представяне в паметта. Сложност на операциите по добавяне, премахване и търсене на елемент. Дефиниране на клас реализиращ двоично кореново дърво за търсене.

Забележка: За изпита ще бъдат избрани две от посочените структури, които да бъдат описани.

Забележка: По този въпрос е възможно да бъдат дадени задачи за практическата част на изпита.

Литература: [11], [17], [23], [26].

16. Софтуерно инженерство и неговото място като дял от знанието. Софтуерен процес и модели на софтуерни процеси. Концепция за многократна употреба.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Софтуерното инженерство – какво е софтуер, видове софтуер, същност и обхват на софтуерното инженерство.
2. Софтуерен процес – фази и основни дейности, видове модели и езици за моделиране, шаблони за описание.
3. Сравнителен анализ на описателни модели на софтуерен процес – модел на водопада, прототипен модел, модел на бързата разработка, спираловиден модел и др.
4. Концепцията за многократна употреба.

Литература: [8], [43], [46].

17. Модели на разпределени софтуерни архитектури. Среда и протоколи за разпределени приложения.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Параметри на паралелната и разпределена обработка, метрика, методи за анализ.
2. Модели на разпределените софтуерни архитектури – процедурни, обектни, потокови, контекстни, йерархични, асинхронни и интерактивни модели на софтуерната архитектура. Структури, организация, компоненти, приложение.

3. Организация на разпределените приложения – клиент-сървер, двуслойни, трислойни и n-слойни модели. р2р. Сървери за приложения и Web-сървери. Метасистеми и грид. Сервизно-, моделно- и аспектино-ориентирани архитектури. Софтуерни агенти.

Примерна задача: Анализ на изчислителната сложност, ускорението, ефективността и цената на обработка на базов или еталонен алгоритъм за паралелна или разпределена обработка.

Литература: [3], [4], [53].

18. Модели и методи за проектиране на потребителски интерфейс.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Основни модели и методи при създаване на потребителски интерфейс: подходи, основни процеси, анализ на задачите, специфициране на взаимодействията, основни техники и инструментални средства.
2. Проектиране на графичен интерфейс: интерактивни стилове и техники, отчитане на психологичните особености на потребителите, концептуални модели и метафори, методи и средства за реализация.
3. Разработка на използваем графичен интерфейс: техники базирани на експерименти.
4. Разработка на мултимедиен графичен интерфейс: проектиране на цветове, звуци, текст, графика, анимация.
5. Особенности при създаване на интегриран интерфейс: методи за моделиране насочени към крайния потребител, екранен дизайн, обработка на взаимодействията, интерактивни методи за проектиране, прототипиране.

Литература: [16], [35], [37], [41], [42], [56].

19. Управление на качеството на софтуерни приложения. Тестване на софтуер.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Осигуряване на качеството – качество на софтуера, алтернативи за осигуряване на качество.
2. Тестови дейности, управление и автоматизация
3. Видове тестване - тестване с контролни списъци, тестване с класове на еквивалентност, разделяне на входния домейн и тестване на границите.
4. Нива на тестване и приложение на техниките за тестване.
5. Измерване. Метрики – характеристики и класификация. Метрики за качество на софтуерен продукт и софтуерен процес.

Примерна задача: Дефиниране на тестови сценарии чрез прилагане на конкретна техника за тестване.

Литература: [9], [34], [36], [48].

20. Софтуерна архитектура. Проектиране и документиране на софтуерни архитектури.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Дефиниция на софтуерна архитектура. Структури и изгледи (*structures and views*) на архитектурата.
2. Изисквания към качеството (нефункционални изисквания) на системата.
3. Проектиране на софтуерната архитектура. Процес за проектиране. Избор на подходящи структури. Последователност на създаване на архитектурата. Тактики (архитектурни решения) за постигане на желаните качествени показатели.

4. Архитектурни стилове.
5. Документиране на софтуерната архитектура. Предназначение на документацията. Основен принцип на документиране. Съдържание на документацията. Структура на документацията.

Примерна задача:

- По зададени изисквания към софтуерната система да се построи модулна декомпозиция на системата.
- По зададена модулна декомпозиция на системата и зададени изисквания да се построи дадена структура

Литература: [29].

21. Инженеринг на софтуерните изисквания. Техники за извличане, анализ и валидиране на софтуерните изисквания. Специфициране на изискванията.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Цел, задачи на инженеринга на софтуерните изисквания.
2. Видове изисквания - класификация
3. Същност на отделните етапи (дейности) на инженеринга на изискванията.
4. Техники за извличане, анализ и валидиране на изискванията, прилагани в отделните етапи на инженеринга на изискванията.
5. Начини за специфициране на изискванията. Видове модели в зависимост от перспективата на системата: модел на потока на данните, модели на поведението, семантични модели, обектно-ориентирани модели, формални модели.

Примерна задача: Извличане и спецификация на изисквания към софтуерна система от примерно описание на такава.

Литература: [38], [47], [55].

22. Проектиране и интегриране на софтуерни системи.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Характеристики на разпределените софтуерни системи – дефиниции, видове системи и тенденции.
2. Междупроцесна комуникация – отдалечено извикване, мултикаст.
3. Разпределени обекти и компоненти.
4. Уеб услуги – дефиниции, шаблони за комуникация. Стандарти за уеб услуги – SOAP, UDDI, WSDL.

Литература: [30], [50].

23. Планиране на проекта – същност и основни елементи, обхват на проекта, времеви и финансови ресурси. Дейности по управление и контрол, методи и средства за създаване на план-график на проекта

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Планиране на проекта – същност и основни елементи на плана на проекта
2. Планиране на обхвата на проекта. Определяне на структура на работа (WBS). План на контролните точки.
3. Планиране на време за изпълнение на задачите, бюджет и необходими ресурси, дейности по управление и контрол
4. Създаване на план - график на проекта. Методи и средства за създаване на план-график. Метод на критичния път. Метод PERT. GANTT диаграми.

Литература: [8], [44].

24. Използване на XML за структуриране, валидация, обработка и представяне на документно съдържание.

Изложението по въпроса трябва да включва следните по-съществени елементи:

1. Добре-структуриран XML – основни концепции, XML йерархии, синтактични правила. XML пространства от имена.
2. XML валидация чрез Document Type Definitions (DTD) – цели на валидирането, DTD структура, синтаксис.
3. XML валидация чрез XML Schema – спецификации, типове данни, фасети, структури. Сравнение с DTD.
4. Използване на XSLT (eXtensible StyleSheet Language Transformations) и XPath за алокиране, манипулиране и представяне на XML съдържание.
5. Използване на DOM (Document Object Model) и SAX (Simple API for XML) за обработка на XML документи – основни интерфейси на DOM и SAX и начини за използването им. Сравнение между DOM и SAX.

Типична задача. По примерно описание на конкретни обекти и връзки между тях, да се състави XML документ и описания на документния му тип с използване на DTD или XML Schema

Литература: [13], [22], [27], [32], [33].

25. Теорема на Ферма. Теорема за средните стойности (Рол, Лагранж и Коши). Формула на Тейлър.

Да се дефинира понятието локален екстремум на функция на една променлива. Да се формулира и докаже необходимо условие за локален екстремум за диференцируеми функции (теорема на Ферма).

Да се докажат следните теореми, формулирани общо за по-кратко.

Нека функцията f е непрекъсната в затворения интервал $[a, b]$ и притежава производна поне в отворения интервал (a, b) . Да се докаже, че:

- а) ако $f(a) = f(b)$, то съществува $c \in (a, b)$ такава, че $f'(c) = 0$ (Рол);
- б) съществува $c \in (a, b)$ такава, че $f(b) - f(a) = f'(c)(b - a)$ (Лагранж);
- в) ако функцията g е непрекъсната в затворения интервал $[a, b]$ и притежава производна поне в отворения интервал (a, b) , като $g'(x) \neq 0$, $x \in (a, b)$, то съществува $c \in (a, b)$ такава, че

$$\frac{f(b) - f(a)}{g(b) - g(a)} = \frac{f'(c)}{g'(c)} \quad (\text{Коши}).$$

По отношение на твърдението във в) да се докаже, че при направените в него предположения имаме $g(a) \neq g(b)$.

За установяването на теоремата на Рол може да се използва без доказателство теоремата на Вайерщрас, според която всяка непрекъсната функция върху краен затворен интервал достига своите максимум и минимум.

Да се изведе формулата на Тейлър с остатъчен член във формата на Лагранж.

Литература: [5], [7], [10].

26. Определен интеграл. Дефиниция и свойства. Интегруемост на непрекъснатите функции. Теорема на Нютон-Лайбниц.

Да се дефинират последователно: разбиване на интервал, големи и малки суми на Дарбу. Да се установи, че при добавяне на нови точки в разбиването на интервала,

големите суми на Дарбу не нарастват, а малките не намаляват (*желателно е да се направи чертеж*). Да се дефинира риманов интеграл чрез подхода на Дарбу.

Да се докаже, че дадена функция е интегрируема по Риман тогава и само тогава, когато за всяко $\varepsilon > 0$ съществуват голяма сума на Дарбу S и малка сума на Дарбу s такива, че $S - s < \varepsilon$. Като се използва тази теорема и теоремата на Кантор (*без доказателство*), според която всяка непрекъсната функция в краен и затворен интервал е равномерно непрекъсната, да се докаже, че всяка непрекъсната функция в краен и затворен интервал е интегрируема по Риман.

Да се изброят (*без доказателство*) основните свойства на римановия интеграл. Да се докаже, че ако f е непрекъсната в $[a, b]$, то съществува $c \in [a, b]$ такава, че

$$\int_a^b f(x)dx = f(c)(b - a).$$

За установяването на това твърдение да се приложат (*без доказателство*) свойството за интегриране на неравенства и теоремата, че всяка непрекъсната функция в $[a, b]$ приема всички стойности между максимума и минимума си.

Да се докаже теоремата на Нютон-Лайбниц, т.е. ако f е непрекъсната в $[a, b]$, то за всяко $x \in [a, b]$ е изпълнено

$$\frac{d}{dx} \int_a^x f(t)dt = f(x).$$

Да се покаже как теоремата се използва за изчисляване на определени интеграли.

Литература: [5], [7], [10].

27. Ранг на система вектори. Ранг на матрица. Теорема за равенството на ранг на матрица, ранг на системата от вектор-редове и ранг на системата от вектор-стълбове на тази матрица. Системи линейни уравнения – теорема на Руше, фундаментална система от решения за хомогенна система и теорема за размерността на подпространството от решения на хомогенна система.

Във въпроса се включва:

- Ранг на система вектори - определение и връзката му с размерността на линейната обвивка; Ранг на матрица – определение; Теорема за равенството на ранг на матрица с ранг на системата от вектор редове и ранг на системата от вектор стълбове на тази матрица (с доказателство);
- Системи линейни уравнения - определение за съвместима и несъвместима система, Теорема на Руше (с доказателство);
- Хомогенни системи – доказателство, че множеството от решения е подпространство, определение за фундаментална система от решения, теорема за размерността на подпространството от решения на хомогенна система (само формулировка).

Литература: [19].

28. Полиноми на една променлива – определение, действия с полиноми и степен на полином. Теорема за делене на полиноми с частно и остатък, правило на Хорнер. Най-голям общ делител на полиноми – твърдение на Безу и алгоритъм на Евклид. Корени на полином и зависимости между корените и коефициентите на полином (формули на Виет).

Във въпроса се включват:

- Определение за полиноми на една променлива с коефициенти над поле, действия с полиномите, степен на полином и твърдение за степента на сумата и на произведението на полиноми (с доказателство);

- Теорема за деление с частно и остатък при полиноми (с доказателство), правило (схема) на Хорнер;
- Най-голям общ делител (НОД) на полиноми с коефициенти от поле - определение, теорема за съществуване на НОД на полиноми и твърдение на Безу (с доказателство), алгоритъм на Евклид за намиране на НОД (с доказателство, че се получава НОД на полиномите);
- Корени на полином – определение, принцип за сравняване на коефициентите (само формулировка), формули на Виет за полином от степен n .

Литература: [18].

29. Дискретни разпределения. Равномерно, биномно, геометрично и Пуасоново разпределение. Задачи, в които възникват. Моменти – математическо очакване и дисперсия.

На изпита комисията дава две разпределения, върху които се развива въпросът. Дефиниция на дискретно вероятностно разпределение на случайна величина. Свойства на вероятностите (неотрицателност и нормираност, монотонност и адитивност). За всяко от дадените две разпределения да се посочи пример, при който то възниква. Да се пресметне математическото очакване и дисперсията на всяко от тези разпределения. При пресмятанятия може да се използва пораждаща функция или пораждаща моментите функция, но тя трябва да се дефинира и да се покажат основните ѝ свойства (без доказателство).

Литература: [6].

ЛИТЕРАТУРА

1. Боровска Пл., Компютърни системи, изд. Сиела, 2005.
2. Боянов Л., К. Боянов и др., Компютърни мрежи и телекомуникации, изд. „Авангард Прима”, София, 2014.
3. Васил Георгиев. Съвременните технологии за конкурентна обработка. Издателство „Св. Климент Охридски”, 2013.
4. Владимир Димитров, Васил Георгиев. Съвременните модели за информационно-технологично обслужване. Издателство „Св. Климент Охридски”, 2012.
5. Джаков П., Р. Леви, Р. Малеев, С. Троянски, Диференциално и интегрално смятане, ФМИ-СУ, София, 2004.
6. Димитров, Б., К. Янев, Вероятности и статистика, Университетско издателство „Св. Климент Охридски”, София, 1998., глави 2.3 (стр. 54-56), 3.2 (стр. 71-74), 6.1 (примери 1-4); [5], тема: Дискретни разпределения.
7. Дойчинов, Д., Математически анализ, Университетско издателство „Св. Климент Охридски”, София, 1994.
8. Илиева С., В. Лилов, И. Манова, Изграждане на софтуерни приложения, 2006, издателство СУ „Св. Климент Охридски”
9. Илиева С., В. Лилов, И. Манова, Методи и подходи за разработване на софтуерни системи, 2010, издателство СУ „Св. Климент Охридски”.
10. Любенова Е., П. Недевски, К. Николов, Л. Николова, В. Попов, Ръководство по Математически анализ, София, 1998.
11. Майерс, С., По-ефективен C++. 35 нови начина да подобрите своите програми и проекти, ЗеСТ Прес.
12. Манев, Кр., Увод в дискретната математика, IV изд., КЛИМН, София, 2005.
13. Маршал Б., XML в примери, , Софт Прес, 2000.
14. Николов Л., Операционни системи, Сиела, София, 1998.
15. Нишева, М., П. Павлов, Функционално програмиране на езика Scheme, София, 2004.
16. Сайт на курса: <http://elearn.uni-sofia.bg/course/view.php?id=22>
17. Седжуик, Р., Алгоритми на С, ч.1-4: Основи, структури от данни, сортиране, търсене, СофтПрес.
18. Сидеров, Пл., Чакърян, К., Записки по алгебра: групи, пръстени, полиноми, Веди, София, 2014
19. Сидеров, Пл., Чакърян, К., Записки по алгебра: линейна алгебра, Веди, София, 2014

20. Тодорова, М., Езици за функционално и логическо програмиране, I ч.: Функционално програмиране. Сиела, София, 2003.
21. Тодорова, М., Обектно-ориентирано програмиране на базата на езика C++, Сиела софт енд пabлишинг АД, 2008.
22. Тодорова, М., Програмиране на C++ I и II част. Siela, София, 2002.
23. Тодорова, М., Структури от данни и програмиране на C++, Сиела Норма АД, 2011.
24. Тодорова, М., Армянов, П., Николов, К., Сборник от задачи по програмиране на C++: част втора, Обектно-ориентирано програмиране, ТехноЛогика, 2008.
25. Тодорова, М., Армянов, П., Петкова, Д., Николов, К., Сборник от задачи по програмиране на C++: част първа, Увод в програмирането, ТехноЛогика, 2008.
26. Уирт, Н., Алгоритми + структури от данни = програми, BG soft group, София.
27. Хънтър Дейвид и колектив, Програмиране с XML, първо издание, Софт Прес, 2000.
28. Abelson, H., & Sussman, G. J. Structure and interpretation of computer programs. The MIT Press, 1996.
29. Brass Len, Paul Clemens, Rick Kazman. Software Architecture in Practice, 2nd Edition, 2003, Addison Wesley.
30. Coulouris G, J. Dollimore, T. Kindberg, G. Blair, Distributed systems – concepts and design, Addison Wesley, 2012
31. Garcia-Molina H., J. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002.
32. Harold E. R., W. S. Means, XML in a Nutshell: A Desktop Quick Reference, O'Reilly & Associates, 3-rd edition, ISBN 0-596-00764-7; 2004
33. Hunter David et al; Beginning XML, 3-rd edition, ISBN-10: 0764570773 , Wrox, 2004
34. Ian Sommerville, Software Engineering, 9th edition (2010), Addison-Wesley Pub Co.
35. Jeff Johnson (2008) GUI Bloopers 2.0: Common User Interface Design Don'ts and Dos (Interactive Technologies) (Paperback) Morgan Kaufman Publishers (Elsevier) ISBN 978-0-12-370643-0
36. Jeff Tian, Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement, 2005, John Wiley & Sons
37. Jenifer Tidwell (2006) Designing Interfaces. Patterns for Effective Interaction Design (Paperback) O'Reilly, ISBN 0-596-00803-1
38. Kotonya Gerald, Ian Sommerville, Requirements Engineering: Processes and techniques, John Wiley&Sons, 2003, ISBN 0 471 97208 8

39. Larry L. Peterson and Bruce S. Davie, Computer Networks: A Systems Approach Fifth Edition, © 2012 Elsevier, Inc.
40. Lewis H., Chr. Papadimitriou, Elements of the theory of computation., Second edition, Prentice-Hall, 1998.
41. Preece, J., Rogers, Y. & Sharp, H. (2002) Interaction Design: Beyond Human-Computer Interaction. New York, NY: John Wiley & Sons. ISBN 978-0-470-01866-8
42. Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S. & Carey, T. (2004). Human-Computer Interaction. Wokingham (GB): Addison-Wesley. ISBN 0130-461091
43. Pressman R., Software Engineering, 2008, Mc Grow Hill
44. Project Management Institute, A Guide to the Project Management Body of Knowledge, <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>
45. Sedgewick Robert, Algorithms in Java, Addison-Wesley, 2002.
46. Sommerville Ian, Software Engineering, 8th edition (2007), Addison-Wesley Pub Co;
47. Sommerville Ian, Software Engineering, Addison Wesley, 8 edition, 2007, ISBN 0 321 31379 8
48. Stephen H. Kan, Metrics and Models in Software Quality Engineering, 2 edition (September 26, 2002) , AddisonWesley Professional.
49. Stroustrup, B., C++ Programming Language. Third Edition, Addison-Wesley, 1997.
50. Tannenbaum Andrew S., Maarten van Steen, Distributed Systems: Principles and Paradigms, Prentice Hall, 2006
51. Tannenbaum, A., Modern Operating Systems, 2nd ed. Prentice Hall, 2002.
52. Tannenbaum Andrew S., Wetherall David J., Computer Networks, 5th ed., Prentice Hall, 2011
53. Tao Lixin, Xiang Fu and Kai Qian. Software Architecture Design - Methodology and Styles. Stipes Publishing L.L.C. 2006.
54. Thompson, S. Haskell: The Craft of Functional Programming (2nd ed.). Addison-Wesley, 1999.
55. Wiegers Karl E., Software Requirements, 2E, ISBN: 0735618798.
56. William Lidwell, Kritina Holden, Jill Butler (2003) Universal Principles of Design. Rockport Publishers Inc, ISBN 1-59253-007-9.