

**СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛ. ОХРИДСКИ”
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА**

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

СПЕЦИАЛНОСТ “СОФТУЕРНО ИНЖЕНЕРСТВО”

приет на ФС № 2/20.02.2012 г.

СОФИЯ • 2012

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) с продължителност 3 астрономически часа. Втория ден изпитът е теоретически (въпроси). Изтегля се 1 теоретичен въпрос и се развива за 2 астрономически часа. След това работите се предават и се прави кратка почивка. Тегли се втори въпрос, който също се развива за 2 часа.

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 (ненадраскано) на актуалния конспект (получава се от отдел „Студенти“ – стая 111). 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. Крайната оценка се формира въз основа на представянето на двете части на изпита. 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. Файлова система. Логическа организация и физическо представяне.
8. Компютърни мрежи и протоколи – OSI модел. Канално ниво. Маршрутизация. IP, TCP, HTTP.
9. Релационен модел. Нормални форми.
10. Процедурно програмиране - основни информационни и алгоритмични структури (на базата на C++).
11. Обектно ориентирано програмиране (на базата на C++ или Java): Основни принципи. Класове и обекти. Конструктори и деструктори. Оператори. Производни класове и наследяване.
12. Структури от данни и алгоритми. Анализ на алгоритми. Абстрактни типове от данни. Стек, опашка, списък, дърво. Сортиране.

СОФТУЕРНО ИНЖЕНЕРСТВО

13. Софтуерно инженерство и неговото място като дял от знанието. Софтуерен процес и модели на софтуерни процеси. Концепция за многократна употреба.
14. Модели на разпределени софтуерни архитектури. Среда и протоколи за разпределени приложения.
15. Модели и методи за проектиране на потребителски интерфейс.
16. Управление на качеството на софтуерни приложения. Тестване на софтуер. План за тестване и тестови сценарии.
17. Софтуерна архитектура. Проектиране и документиране на софтуерни архитектури.

18. Инженеринг на изискванията като част от системния и софтуерен инженеринг. Представяне на техники, прилагани на отделните етапи от инженеринга на софтуерните изисквания. Основни методи за моделиране.
19. Интеграция на разпределени информационни системи.
20. Планиране на проекта – обхват на проекта, структурата на работа по проекта, време за изпълнение на задачите и създаване на план – график, планиране на ресурси, разходи и бюджет на проекта.
21. Използване на XML за структуриране, валидация, обработка и представяне на документно съдържание.

МАТЕМАТИКА И ПРИЛОЖЕНИЯ

22. Теорема за средните стойности (Рол, Лагранж и Коши). Формула на Тейлър.
23. Определен интеграл. Дефиниция и свойства. Интегруемост на непрекъснати функции. Теорема на Нютон-Лайбниц.
24. Ранг на матрица. Ранг на система вектори. Теорема за ранга. Системи линейни уравнения – Теорема на Руше, фундаментална система от решения на хомогенна система.
25. Дискретни разпределения. 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:

Манев, Кр., *Увод в дискретната математика*, IV изд., КЛИМН, София, 2005.

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:

Манев, Кр., *Увод в дискретната математика*, IV изд., КЛИМН, София, 2005.

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 (uvw) (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:

1. Манев, Кр., *Увод в дискретната математика*, IV изд., КЛИМН, София, 2005.
2. Papadimitriou, *Elements of the theory of computation.*, Second edition, Prentice-Hall, 1998.

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

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

Литература:

1. Манев, Кр., *Увод в дискретната математика*, IV изд., КЛИМН, София, 2005.
2. Lewis H., Chr. Papadimitriou, *Elements of the theory of computation.*, Second edition, Prentice-Hall, 1998.

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.

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:

Боровска Пл., *Компютърни системи*, изд.Сиела, 2005.

7. Файлова система. Логическа организация и физическо представяне.

1. Логическа организация на файлова система:

- имена на файлове.
- типове файлове - обикновен файл, специален файл, каталог, символна връзка, програмен канал.
- вътрешна структура на файл.
- атрибути на файл.
- йерархична организация на файлова система - абсолютно и относително пълно име на файл, текущ каталог.

2. Физическо представяне на файлова система:

- Изисквания и способности за управление на дисковите области.
- Примери за физическа организация на файлова система:
 - UNIX Sysyem V;
 - LINUX;
 - MSDOS;
 - NTFS.

Забележка: За изпита ще бъдат избрани два от изброените примери за файлова организация.

Литература

1. Tanenbaum, A., *Modern Operating Systems*, 2nd ed. Prentice Hall, 2002.

2. Николов Л., *Операционни системи*, Сиела, София, 1998.

8. Компютърни мрежи и протоколи – OSI модел. Канално ниво. Маршрутизация. IP, TCP, HTTP.

- 1) Седемслоен модел OSI на ISO – характеристики на нивата.
- 2) Канално ниво – кадри, предаване, грешки, номерация, прозорци.
- 3) Метод на достъп до съобщителната среда в ЕТЕРНЕТ и формат на кадрите .
- 4) Разпределена маршрутизация: алгоритъм с дистантен вектор и алгоритъм със следене състоянието на връзката. IP протокол – формат на дейтаграмата,

адресация, подмрежи и маски. TCP протокол и установяване на съединения. Хипертекстов протокол HTTP.

Литература:

1. Tannenbaum A., *Computer Networks*- 3th ed., 4th ed., Prentice Hall
2. Комър Браян, *TCP/IP мрежи и администриране*, изд.ИнфоДар, 1999
3. Боянов К., Хр. Турлаков, Дим. Тодоров, Л. Боянов, Вл. Димитров, Вед. Желязков – *Принципи на работа на компютърните мрежи*. ИНТЕРНЕТ. изд. Апиинфоцентър Котларски, 2003.

9. Релационен модел. Нормални форми.

1. Релационен модел на данните: домен; релация; кортеж; 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.

2. 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.

Забележка. На изпита ще бъде давана една от двете части на въпроса.

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

Литература:

1. Азълов, П., *Бази от данни. Релационен и обектен подход*, Техника, София, 1991
2. Garcia-Molina H., J. Ullman, J.Widom, *Database Systems: The Complete Book*, Prentice Hall, 2002.

10. Процедурно програмиране - основни информационни и алгоритмични структури (на базата на C++).

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

1. Скаларни типове от данни. Логически тип. Числени типове цял и реален.
2. Съставни типове от данни. Структура от данни масив. Тип масив.
3. Тип указател – дефиниране, основни операции. Указателна аритметика.
4. Указатели и едномерни масиви. Указатели и двумерни масиви. Указатели и низове.
5. Функции. Дефиниране на функция. Обръщение към функция.
6. Предаване на параметрите по стойност, чрез указател и чрез псевдоним.

7. Функции. Масиви като формални параметри.

Типична задача. Да се състави функция, която въз основа на зададени като параметри масиви и/или матрици, чрез съответен анализ формира други такива.

Литература:

1. Тодорова, М., *Програмиране на C++*. I и II част. Ciela, София, 2002.
2. Stroustrup, B., *C++ Programming Language*. Third Edition, Addison-Wesley, 1997.

11. Обектно ориентирано програмиране (на базата на C++ или Java): Основни принципи. Класове и обекти. Конструктори и деструктори. Оператори. Производни класове и наследяване.

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

1. Основни принципи на обектно ориентираното програмиране.
2. Класове. Дефиниране на класове. Област на класове. Обекти.
3. Конструктори. Дефиниране на конструктори. Видове конструктори.
4. Указатели към обекти. Масиви и обекти. Динамични обекти.
5. Деструктори. Създаване и разрушаване на обекти на класове.
6. Оператори. Предефиниране на оператори.
7. Производни класове. Дефиниране. Наследяване и достъп до наследените компоненти.

Литература:

1. Тодорова, М., *Програмиране на C++*. I и II част. Ciela, София, 2002.
2. Stroustrup, B., *C++ Programming Language*. Third Edition, Addison-Wesley, 1997.

12. Структури от данни и алгоритми. Анализ на алгоритми. Абстрактни типове от данни. Стек, опашка, списък, дърво. Сортиране.

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

1. Основи на анализа на алгоритми: Асимптотична нотация на сложността. Основни рекурентни формули. Примери за анализ на алгоритми.
2. Абстрактни типове от данни. Интерфейс и реализация.
3. Свързани списъци. Обработка на списъци.
4. Структура от данни стек Реализация.
5. Структура от данни опашка. Реализация
6. Дървета. Типове дървета.
7. Сортиране. Елементарни методи за сортиране.
8. Сортиране - QuickSort.

Литература

Sedgewick Robert, *Algorithms in Java*, Addison-Wesley, 2002.

13. 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:

1. Илиева С., В. Лилов, И. Манова, *Изграждане на софтуерни приложения*, 2006, издателство СУ “Кл. Охридски”
2. Sommerville Ian, *Software Engineering*, 8th edition (2007), Addison-Wesley Pub Co;
3. Pressman R., *Software Engineering*, 2008, Mc Grow Hill

14. 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-сервер, двуслойни, трислойни и n-слойни модели. р2р. Сервери за приложения и Web-сервери. Метасистеми и грид. Сервизно-, моделно- и аспектно-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:

1. Tao Lixin, Xiang Fu and Kai Qian. *Software Architecture Design - Methodology and Styles*. Stipes Publishing L.L.C. 2006.
2. Владимир Димитров, Васил Георгиев. *Съвременните модели за информационно-технологично обслужване*. Издателство „Св. Кл. Охридски”, 2012.
3. Васил Георгиев. *Съвременните технологии за конкурентна обработка*. Издателство „Св. Кл. Охридски”, 2013.

15. 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:

1. Сайт на курса: <http://elearn.uni-sofia.bg/course/view.php?id=22>
2. 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
3. Preece, J.; Rogers, Y.; Sharp, H.; Benyon, D.; Holland, S. & Carey, T. (2004). *Human-Computer Interaction*. Wokingham (GB): Addison-Wesley. ISBN 0130-461091
4. William Lidwell, Kritina Holden, Jill Butler (2003) *Universal Principles of Design*. Rockport Publishers Inc, ISBN 1-59253-007-9
5. Jenifer Tidwell (2006) *Designing Interfaces. Patterns for Effective Interaction Design* (Paperback) O'Reily, ISBN 0-596-00803-1
6. 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

16. Управление на качеството на софтуерни приложения. Тестване на софтуер. План за тестване и тестови сценарии.

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

1. Модели на качеството (класически и съвременни).
2. Класификация и видове тестове в зависимост от готовността на софтуерното решение - тестване на софтуерна единица, тестване на модул, интеграционно тестване на модули, тестване на система, потребителски тестове за приемане на системата).
3. Подходи за тестване - бяла кутия, черна кутия, сива кутия.
4. Класификация и видове тестове, свързани с функционални и технически изисквания към системата - функционално, регресионно, тестване на производителност, тестване на сигурност, други.
5. Техники за оптимизация на функционални тестови сценарии - клас на еквивалентност, анализ на граничните стойности, таблица за вземане на решение).
6. План за тестване и тестови сценарии – предназначение и примерна структура

Примерна задача: Да се създадат тестови сценарии по зададена примерна софтуерна система

Литература:

1. Илиева С., В. Лилов, И. Манова, *Изграждане на софтуерни приложения*, 2006, издателство СУ “Кл. Охридски”
2. Horch John, *Practical guide to software quality management*, second edition, Artech House,
3. Galin D., *Software Quality Assurance : From Theory to Implementation*, (2008), Pearson Addison-Wesley

17. 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:

Brass Len, Paul Clemens, Rick Kazman. *Software Architecture in Practice*, 2nd Edition, 2003, Addison Wesley.

18. Инженеринг на изискванията като част от системния и софтуерен инженеринг. Представяне на техники, прилагани на отделните етапи от инженеринга на софтуерните изисквания. Основни методи за моделиране.

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

- 1) Цел и задачи на инженеринга на софтуерните изисквания.
- 2) Параметри на процеса – видове изисквания, класификация, потребители.
- 3) Етапи (дейности) на инженеринга на изискванията.
- 4) Техники за извличане, анализ и валидиране на изискванията.
- 5) Основни методи за моделиране – Data-flow модели, семантични модели, обектно-ориентирани модели, формални модели.

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

Литература:

1. Sommerville Ian, *Software Engineering*, Addison Wesley, 8 edition, 2007, ISBN 0 321 31379 8
2. Kotonya Gerald, Ian Sommerville, *Requirements Engineering: Processes and techniques*, John Wiley&Sons, 2003, ISBN 0 471 97208 8
3. Wiegers Karl E., *Software Requirements*, 2E, ISBN: 0735618798.

19. Интеграция на разпределени информационни системи.

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

- 1) Типове мидълуер и начини на използването им при интеграция на софтуерни приложения;
- 2) Проблеми на интеграцията на софтуерни приложения с използването на обмен на файлове, по сокети, RPC и RMI;
- 3) Интеграция на Уеб приложения и технологии за поддръжка на отдалечени клиенти;

- 4) Интеграция с използването на Уеб услуги - технологии за описание, откриване, взаимодействие и композиция на услуги.

Литература:

1. Alonso Gustavo, etc., *Web services – Concepts, Architecture and Applications*, Springer-Verlag, 2004
2. Tanenbaum Andrew S., Maarten van Steen, *Distributed Systems: Principles and Paradigms*, Prentice Hall, 2006
3. Erl Thomas, *SOA: Principles of Service Design*, Prentice Hall, 1 edition, 2007

20. Планиране на проекта – обхват на проекта, структурата на работа по проекта, време за изпълнение на задачите и създаване на план – график, планиране на ресурси, разходи и бюджет на проекта.

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

1. Планиране на проекта – обхват на проекта и определяне на структурата на работа по проекта (WBS)
2. Планиране на проекта – време за изпълнение на задачите, мрежови диаграми
3. Методи и средства за създаване на график. Критичен път. Метод на критичния път. Метод PERT. GANTT диаграми. Създаване на план - график на проекта
4. Планиране на проекта – ресурси, разходи и бюджет на проекта

Литература:

1. Илиева С., В. Лилов, И. Манова, *Изграждане на софтуерни приложения*, 2006, издателство СУ “Кл. Охридски”
2. Heerkins G.R., *Project Management*, McGraw–Hill, 2002
3. Keeling R., *Project Management an International Perspective*, Macmillan Press, 2000

21. 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:

1. Hunter David at al; *Beginning XML*, 3-rd edition, ISBN-10: 0764570773 , Wrox, 2004
2. 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
3. Хънтър Дейвид и колектив, *Програмиране с XML*, първо издание, Софт Прес, 2000.
4. Маршал Б., *XML в примери*, , Софт Прес, 2000.
5. Тодорова, М., *Програмиране на C++ I и II част*. Ciela, София, 2002.

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

Необходимо е да се докажат следните, формулирани общо, теореми: Нека f е непрекъснатата 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 такова $c \in (a, b)$, че $f'(c) = 0$ (**Рол**);
- б) съществува такова $c \in (a, b)$, че $f(b) - f(a) = f'(c)(b - a)$ (**Лагранж**);
- в) ако g е непрекъснатата в затворения интервал $[a, b]$ и притежава производна поне в отворения интервал (a, b) , $g'(x) \neq 0, x \in (a, b)$, то съществува такова $c \in (a, b)$, че

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

За доказателството на теоремата на Рол да се използва (*без доказателство!*) теоремата на Вайерщрас, според която всяка непрекъсната функция в краен и затворен интервал достига своя максимум и минимум.

Необходимо е още да се изведе формулата на Тейлър с остатъчен член във формата на Лагранж.

Литература:

1. Джаков П., Р. Леви, Р. Малеев, С. Троянски, *Диференциално и интегрално смятане*, ФМИ-СУ, София, 2004.
2. Дойчинов, Д., *Математически анализ*, Университетско издателство “Св. Кл. Охридски”, София, 1994.

3. Любенова Е., П. Недевски, К. Николов, Л. Николова, В. Попов, *Ръководство по Математически анализ*, София, 1998.

23. Определен интеграл. Дефиниция и свойства. Интегруемост на непрекъснати функции. Теорема на Нютон-Лайбниц.

Да се дефинират последователно: разбиване на интервал, диаметър на разбиване, риманова сума и риманов интеграл. Да се покаже, че всяка интегруема по Риман функция е ограничена.

Да се дефинират 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 докаже, че дадена функция е интегруема 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, 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 интегруема по Риман. Да се изброят (*без доказателство*) основните свойства на Римановия интеграл. Като се приложи свойството за интегриране на неравенства и теоремата, че всяка непрекъсната функция приема всички стойности между максимума и минимума си, да се докаже, че ако f е непрекъсната в $[a, b]$, то съществува $c \in [a, b]$ така, че

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

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

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

да се покаже как теоремата се използва за изчисляване на определен интеграл.

Литература:

1. Джаков П., Р. Леви, Р. Малеев, С. Троянски, *Диференциално и интегрално смятане*, ФМИ-СУ, София, 2004.
2. Дойчинов, Д., *Математически анализ*, Университетско издателство “Св. Кл. Охридски”, София, 1994.
3. Любенова Е., П. Недевски, К. Николов, Л. Николова, В. Попов, *Ръководство по Математически анализ*, София, 1998.

24. Ранг на матрица. Ранг на система вектори. Теорема за ранга. Системи линейни уравнения – Теорема на Руше, фундаментална система от решения на хомогенна система.

Във въпроса се включва – определение за ранг на система вектори и ранг на матрица, теорема за равенство на ранга на системата вектор - редове и вектор-стълбове на една матрица. Системи линейни уравнения – съвместими, несъвместими и теорема на Руше. Хомогенна система, фундаментална система от решения – теорема за брой решения във фундаментална система от решения, връзка между решенията на хомогенна система и решения на нехомогенна система.

Литература:

Сидеров, Пл., *Записки по алгебра: линейна алгебра*, Веди, София, 2001.

25. Дискретни разпределения. Задачи, в които възникват. Моменти – математическо очакване и дисперсия.

Дефиниция на (дискретно и) целочислено разпределение на случайна величина. Свойства на вероятностите (неотрицателност и нормираност). За всяко от разпределенията – равномерно, биномно, геометрично, Пуасоново и хипергеометрично – да се посочи пример (задача), при който то възниква. Пресмятане на математическото очакване и дисперсията на всяко от тези разпределения. При пресмятанията може да се използва пораждаща моментите функция, но тя трябва да се определи за всяко целочислено разпределение и да се изведат основните й свойства.

Литература:

Димитров, Б., К. Янев, *Вероятности и статистика*, Университетско издателство “Св. Кл.Охридски”, София, 1998., глави 2.3 (стр. 54-56), 3.2 (стр. 71-74), 6.1 (примери 1-4); [5], тема: Дискретни разпределения.