Софийски Университет "Климент Охридски"

Структури от данни и програмиране

Едносеместриален задължителен курс (4+4) за студенти от
ФМИ (математика и информатика, математика)
ФзФ (физика и информатика) и

Конспект

Преподавател:
доцент д- р Евгений Кръстев, e-mail: eck@fmi.uni-sofia.bg

Хорариум: 4 лекции + 4 (практически занятия в компютърна зала)

Специалност и курс:
студенти от втори курс, трети семестър от степен бакалавър от специалност математика и информатика и специалност математика (ФМИ**) и специалност Физика и Информатика** (ФзФ)

Анотация

Основната **цел** на курса е студентите да овладеят **класическите техники за структуриране на данни** и **програмната реализация** на структури от данни **в типични алгоритми с приложения** в информатиката, математиката и химията. С оглед на постигане на тази цел, **тематиката на курса фокусира върху практическото приложение на принципите за обектно ориентираното програмиране** (ООП) с използване на езика за програмиране ***Java***. Основните теми на курса включват изучаване на **класически структури от данни (***масиви, данни от изброим тип, файлове и сериализация,вътрешни класове и обработка на събития, свързани списъци, стек, опашки, дървовидни структури, графи и др***.), типични алгоритми** *(сортиране и търсене, статистически операции),* илюстриращи приложението на тези структури от данни, **както и програмна реализация на езика *Java*** *(реализация на графичен потребителски интерфейс с мултимедиини възможности, параметризация типове данни при работа с методи и класове, както и стандартни библиотеки за работа със структури от данни)* **на тези алгоритми**. Акцентира се върху придобиване на **добър стил** за програмиране.

По време на курса, принципите на ООП се представят посредством **пълни и реалистични програмни приложения** на ***Java*** като се използват предимствата на ***J2SE 7***. Курсът се съпровожда с подробни ***Power Point*** лекции за представяне на разглеждани теми, от множество примерни програмни решения, упражнения и курсов проект, които позволяват да се изясни решаването на често срещани бизнес проблеми в предвидените практически занятия. **Материалите по курса са достъпни *on- line*** в рамките на специализирана **система за електронно обучение**, която служи за **стимулиране и оценка на индивидуалната работа** на студентите.

Забележка: Практикумът „***Структури от данни и програмиране***" (0+0+2) за специалност ***Математика*** (ФМИ) е задължителен и се провежда заедно с курса „***Структури от данни и програмиране***" (4+0+2) за специалност ***Математика*** (ФМИ) като се следва програмата на

курса „***Структури от данни и програмиране***" (4+0+4) за специалност ***Математика и Информатика***(ФМИ).

Провеждане на курса и изисквания към студентите

Курсът се предлага през **зимния семестър** (октомври/ януари) и е предназначен за **студенти от втори курс степен бакалавър**. Необходимо **изискване за този курс** е успешното приключване на курса "***Увод в програмирането***".. Допуска се също, че студентите имат **задоволителни знания по английски език**, позволяващо им да ползват литература в областта на информатиката.

Конспект- основни теми

1. **Масиви**- едномерни и многомерни. Дефиниция и типични проложения- **програмна реализация на алгоритми** с приложения на масиви.
2. **Структуриране на данни с** `enum` **типове данни- програмна реализация** на алгоритми с приложения на `enum` **типове данни.**
3. **Побитови операции** в **Java- оператори за** по битови ***AND, OR, right shift, left shift, complement exclusive OR*** . Дефиниция и основни приложения, алгоритми.
4. **Вътрешни и анонимни класове-** синтаксис и приложения. ***public*** и ***private*** конструктори, скриване на source кода на приложението
5. **Обработка на събития** с вътрешни и анонимни класове (***closure , callback*** *конструкции*)
6. **Програмиране на събития с използване на вътрешни и анонимни класове**. Обработка на **събития породени от мишка и клавиатура**.
7. **Създаване на графичен потребителски интерфейс**. Моделиране и програмна реализация на потребителския интерфейс с ***NetBeans*** **с използване на вътрешни и анонимни класове за обработка на събития.**
8. **Низове, символи и регулярни изрази**- класове ***String, Character, StringBuilder*** и ***StringTokenizer***.
9. **Създаване на шаблони и обработка на регулярни изрази** ***class Pattern***, ***class Matcher***. **Проверка за валидност** на потребителския вход.
10. **Програмна реализация** на **типични алгоритми за** сортиране. Алгоритми и програмна реализация на методи за сортиране- ***selection***, ***insertion*** и ***counting*** сортиране. Оценки за **ефективността на алгоритми** за сортиране**.**
11. **Обработка на изключения.** Класът ***Exception***. ***Try*** блок и приложението му. Прихващане и предаване на изключение.
12. **Потребителски дефинирани изключения** с използване на наследственост. Приложения- ***assertions***.
13. **Файлове и потоци от данни.** Клас ***File*** и приложения. Създаване, четене и редактиране на файл с последователен достъп(***InputStream/***

OutputStream- буфериран и не- буфериран вход и изход, ***Reader/ Writer*** и аналогични класове.

14. ***Serializable*** обекти. Създаване на **повторно използваеми графични интерфейси**- концепция и приложението ѝ при обработка на файл. Диалогови прозорци за избор на файл.

15. **Model- View- Controller модел** за изграждане на приложения. Примери за **приложение на MVC модела** при обработка на **файл със сериализиране** на обекти.

16. **Файлове с произволен достъп**. Примери за приложение на MVC модела **обработка на файл с произволен достъп**- методи за четене и писане, метод `seek()`. Решение на **задачата за обработка на транзакции** с използване на MVC модела

17. **Структури от данни.** Класове за представяне на примитивни данни- ***autoboxing /auto- unboxing***. Рекурсивно дефинирани класове. Динамично заделяне на памет.

18. **Класически типове данни**- свързани списъци (***linked lists***) Видове. Дефиниция

19. **Свързани списъци**- типични **алгоритми** и **програмна реализация**

20. .**Класически типове данни**- стек (***stack***) Видове,. Дефиниция

21. **Рекурсия**. Примери за илюстриране на концепцията. Рекурсия и стек на извиканите методи. Рекурсия и итерация. Рекурсивно връщане (***backtracking***). **Приложения на рекурсия- алгоритми за** сортиране сортиране- ***selection***, ***insertion*** и ***merge*** , свързани списъци, бинарни дървета.

22. **Приложение на стек** за анализ и пресмятане на ариметичен израз. Разделяне на израза на лексеми, представяне в обратен полски запис

23. **Класически типове данни**- опашки (***queues***) Видове, дефиниция, типични алгоритми и програмна реализация.

24. **Класически типове данни**- дървета(***trees***) Видове,. дефиниция, типични алгоритми и програмна реализация. Представяне на двоично дърво за търсене. Обхождане на бинарно дърво за търсене- ***preorder, inorder u postorder***.

25. **Основни операции върху двоично дърво**. «Червено- Черно» дърво (red-black trees) като разновидност на двоично дърво с гарантирано балансиране.

26. **Граф**. Представяне на граф чрез **списъци на инцидентност.** Обхождане на граф в ширина (**BFS**). Алгоритми за намиране на най-кратък път - като брой дъги и като сума от дължината на дъгите (алгоритъм на ***Dijkstra***)

27. **Обхождане на граф** в дълбочина (***DFS***). Топологично сортиране

28. **Пораждане на методи** (***generics***). Пораждащи методи- приложения и особености при компилиране. Презареждане (**overloading**) на пораждащ метод.

29. **Пораждане на класове** (***generics***). Първичен (raw) вид на пораждащ клас. Шаблони (неявни аргументи, ***wildcards***) за тип в методи които използват аргументи.

30. **Приложения на параметризирани** класове и методи. **Основни операции върху двоично дърво**. *Heap*, *Priority queue* и *HeapSort* - алгоритми и програмна реализация с параметри за тип.

31. Съвъкупности от данни (***collections***). Клас ***Arrays***, ***interface Collection*** и ***class Collections***. Използване на интерфейс ***Iterator***. Примери за списъци- ***ArrayList*** , ***LinkedList*** и ***Vector***.

32. **Структури от данни- множества** (***Set***) и **карти**(***Map***) и приложения.

33. **Алгоритми и програмна реализация** използващи съвъкупности от данни- ***sort***, ***shuffle, reverse, fill, copy, max, min, binarySearch, frequency*** и други.

34. **Мултимедия- аплети и приложения.** Зареждане, изобразяване и скалиране на изображения Анимиране на серия от изображения. Зареждане и изпълнение на музикални и видео клипове с **Java Media Framework**

35. **Въведение в JavaFX.** Декларативен синтакс. Представяне на обекти от графичния интерфейс в графична сцена и примерна програмна реализация на мултимедийни приложения с **JavaFX**.

Забележка: ***Задачите*** представени като част от упражненията за илюстрация на лекциите са ***неделима част от този конспект*** и тяхните решения ще ***се изискват в края на всяко практическо занятие.***.

Изпити и оценки

Окончателната оценка се базира на **два писмени изпита** (всеки с 20% тежест в крайната оценка) по *време на курса*, ***финален*** **писмен изпит** (40%) и защита на три от **домашните работи** (20%), предадени след първото задължително контролно.
Оценката от защитата на домашните включва освен оценката от конкретното домашно, също и мнението на асистента за работата студента през семестъра и присъствието на студента в практическите занятия.
Двата **писмени изпита** по *време на курса и* **защитата на домашните работи** не се повтарят **след приключване на семестъра.**
Оценките се образуват на базата на точките, съответстващи на процента от изпълнението на съответното задание (контролно или защита на домашни) по следната скала:

Оценки:

2	**от 0 до 54 точки**
3	**от 55 до 64 точки**
4	**от 65 до 74 точки**
5	**от 75 до 84 точки**
6	**от 85 до 100 точки**

Забележка: **В случай, че в края на семестъра средната оценка от контролните и домашните, взети със съответните тежести, е по- малка от 55 точки, то студентът трябва да издържи писмен поправителен изпит с поне 65 точки, за да завърши курса**.

Литература

[1] **P. Deitel, H. Deitel, "**Java How to Program (early objects)**",** Prentice Hall 9 ed. **2011**, ISBN-10: 0132575663, ISBN-13: 978-0-13-257566-9 (**основна)**

[2] **Y. Daniel Liang, "**Introduction to Java Programming**", 7th** ed.,Prentice Hall **2009** ISBN-13: 978-0-13-601267-2

[3] **Gail Anderson, Paul Anderson** "Essential JavaFX", Prentice Hall **2009** ISBN-10: 0-13-704279-5

[4] **T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein,** "Introduction to Algorithms", 2nd Edition, The MIT Press, **2001, ISBN** 0-262-53196-8

[5] **Peter Drake**, "Data Structures and Algorithms in Java", Prentice Hall, **2005, ISBN-10**: 0-13-146914-2

[6] **Bruce Eckel** "Thinking in Java", 4th ed., Prentice Hall 2006
или българското й издание "Да мислим на Java" том 1 и 2, SoftPress, **2001**

Септември 2012