

У в о д в П р о г р а м и р а н е т о

Едносеместриален задължителен курс (4+4) за студенти от

ФМИ (математика и информатика)

ХФ (химия и информатика) и

ФзФ (Физика и Информатика)

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 първи курс, втори семестър от степен бакалавър от специалност математика и информатика (ФМИ), специалност химия и информатика (ХФ) и специалност Физика и Информатика (ФзФ)

Анотация

Курсът е предназначен за студенти без първоначални знания и умения в областта на програмирането. Основната цел на курса е студентите да овладеят **език за програмиране с широко приложение в софтуерната индустрия**, което в края на курса да им позволи да осъществяват **алгоритмична и програмна реализация на типични задачи** от областта на информатиката, математиката и химията, както и да служи за основа на професионалната им реализация в бъдеще. 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 **основни алгоритми и тяхната визуализация**, изучаване на **основите на езика Java**, позволяващо **програмна реализация** на типични алгоритми с използване на **наследственост и полиморфизъм**, както и **разработка на графичен потребителски интерфейс** с помощта на **Swing** графични компоненти. Акцентираща 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 6.0**. 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.

Провеждане на курса и изисквания към студентите

Курсът се предлага през **летния семестър** (февруари/ юни) и е предназначен за **студенти от първи курс степен бакалавър**. Общи познания за работа с **MS Windows** са **препоръчителни** изисквания за този курс. Допуска се също, че студентите имат

Софийски Университет “Климент Охридски”

задоволителни знания по английски език, позволяващо им да ползват литература в областта на информатиката.

Конспект- основни теми

1. Въведение в програмирането. Видове **данни и цифрово представяне - цифрови системи и типични операции с данни.**
2. **Основни понятия от областта на операционните системи.** Въведение към Компютърни системи и Програмни езици. Въведение към ООА и ООД
3. **Инсталиране на Java и изпълнение на Java програми.** Програмни езици и среди за разработка (Integrated Development Environment- IDE). **Стил на програмиране.** Инсталиране на **JDK 6**, **JDK** документация и **NetBeans**. Основни **стъпки в създаване на приложения на Java**
4. Приложения на **обектно ориентирания анализ** при дефиниране на класове, обекти- **статус и поведение.** Първа програма на **Java**- ключови думи, основни правила на синтаксиса и препоръки за добър стил на програмиране. **Компилиране и изпълнение.**
5. Създаване на **Java Console приложения** (вход/изход с **J2SE 6.0- class Scanner, System.out.printf()** за форматиране на изходни данни). Създаване и редактиране на **Java** приложения. **Представяне на данни в паметта- типове данни.**
6. **Аритметични операции** и въведение в **управление на логиката на изпълнение**- оператор **if** и оператори за сравнение. **Графичен вход и изход с диалогов прозорец.**
7. **Въведение към ООП-** класове и обекти. **Основни термини** в обектно ориентираното програмиране. Капсулиране на данни и методи, които оперират с тези данни.
8. **Въведение към UML.** Деклариране на класове в **Java**- идентификатори, примитивни и референтни типове данни, методи и свойства- **set** методи и **get** методи. Създаване на обекти от даден клас.
9. **Инициализиране на обекти с конструктори-** видове. Използване на данни с плаваща запетая. **Създаване на форматиран String- метод format(), типове данни за числа с плаваща запетая, явно преобразуване на типове данни, форматиращи спецификатори.**
10. **Алгоритми, псевдокод и визуализирането му- I част.** Елементи на **структурно програмиране- управляващи структури**, визуализацията им и тяхните аналози в **Java, синтаксис**. за **if** и **while**
11. **Основни алгоритмични постановки**, видове цикли- псевдокод и програмна реализация. Оператори за инкрементиране и декрементиране. Използване на графичен потребителски интерфейс (**GUI**) за създаване на **прости графични изображения.**
12. **Алгоритми, псевдокод и визуализирането му II част.** Елементи на **структурно програмиране- управляващи структури**, компютърни методи за визуализацията им и тяхните аналози в **Java, синтаксис**. за **for, switch, break, continue** - типични алгоритмични **приложения** и програмна реализация
13. **Логически оператори.** Преобразуване на **String** към други типове примитивни типове данни (**parseInt(), parseDouble()** etc). **Swing компоненти-** приложения с **JOptionPane.showMessageDialog()** за изобразяване на текст и

Софийски Университет "Климент Охридски"

`JOptionPane.showInputDialog()` за четене на потребителския вход. Използване на графичен потребителски интерфейс (*GUI*) за рисуване на правоъгълници и елипси.

14. **Особености в използването на методи в Java. Програмни модули** - клас, метод и package. Приложение на принципа "*разделяй и владей*". Статични методи и данни `class Math` и методи. Съчленяване на низове.
15. Особености в декларирането и използването на статични методи. Машино представяне на изпълнението на методи- *stack* структура и *LIFO* организация на данни, "*execution stack*", "*activation record*".
16. Предаване на данни между методи- *stack* и *heap* структури, предназначение. **Разлика в представяне на примитивни и референтни данни в паметта**, организация на *stack* и *heap* на процес. Примери за използване на *stack* и *heap* при копиране на данни- случай на примитивни и референтни данни. **Сравнение между указатели и референтни променливи.**
17. Генериране на случайни числа. Класът *Random* и приложения на неговите методи. **Обсег на валидност на декларации на данни.** Допълнително **дефиниране на методи (*overloading*)**. GUI приложения- използване на `class Color`.
18. Масиви- деклариране и дефиниране, методи и свойства на масиви, примерни приложения. Приложения на масиви- сумиране на елементи, статистически анализи (броене на честота на признак)
19. Специализиран цикъл `for` за **обработка на масив** и списък от **аргументи с променлива дължина (J2SE 6.0)**. **Предаване на аргументи на методи.** Предаване на масиви като аргументи.
20. Предаване на данни по стойност и предаване на данни по референция- общ случай и конкретна реализация в Java.
21. **Многомерни масиви и приложения**- методи с променлив брой аргументи, използване на аргументи от кода на ред. Приложения в GUI.
22. Търсене и сортиране на масив. Алгоритъм и програмна реализация на последователно и бинарно търсене.
23. Алгоритъм и програмна реализация на методи за сортиране- "*метод на мехурчето*". Примери за използване на графични компоненти за построяване на базисен графичен интерфейс- извеждане на текстови данни.
24. **Обектно ориентирано програмиране** - данни и методи на клас, дефиниране на конструктори за общо ползване, по подразбиране и копиране. Ролята на методи от тип `accessor (get)` и `mutator (set)`.
25. Управление на достъпа до данни и методи на клас- *public*, *private*, *package* достъп. "**Overloading**" на конструктори - референцията *this*. Правила за писане на класове
26. *final* данни на клас. Обекти като данни на клас- **композиция**. Статични данни и методи. **Инициализиране на статични данни.** Метода `finalize()` и "*garbage collection*". приложение при броене на текущия брой обекти от даден клас. Използване на библиотеки със *static import* и *enum* типове данни в J2SE 6.0. **Създаване на потребителски библиотеки в Java.** Използване на `class Graphics`
27. **Наследственост**- базов клас и производни класове. Многократно използване на софтуер "*software reusability*". *protected* данни и методи. Конструктори и `finalize()` при производните класове- особености.

Софийски Университет "Климент Охридски"

28. **Правила за писане на класове в йерархия от наследственост**- използване на ключовата дума *super* за извикване на конструктора на базовия клас, Методи на *class Object*. Използване на *Swing* компоненти- *JLabel*. Извеждане на текст и картина и позициониране с *BorderLayout*
29. **Полиморфизъм**- късно свързване и приложението му при Java. **Абстрактни** класове и методи. "*Overriding*" на методи.
30. Приложение на полиморфизъм. **Преобразуване нагоре и преобразуване надолу**. Приложение на оператор *instanceof* .
31. **Интерфейси**- синтаксис и приложения при **множествено онаследяване**. **UML представяне** на интерфейси в релации на наследственост. **final** клас, **final** метод
32. Създаване на **графичен потребителски интерфейс**. Основни *Swing* компоненти и **обработка на събития** в GUI. **Моделиране и програмна реализация на потребителския интерфейс с NetBeans**. Обработка на **събития** породени от мишка и клавиатура.
33. Адаптер класове и управление на подредбата на графичните компоненти *Layout managers*. - получаване на референция към графичния контекст и добавяне на графични компоненти (*getContentPane()*, *setLayout()* на клас *Container*). Приложения.
34. **Синтаксис и приложения при обработка на събития**. Модел за обработка на събитията и приложение на полиморфизъм при реализацията на модела. Приложения с *ActionListener*, *MouseListener*, *MouseMotionListener* и *KeyListener* , както и съответните им събития.
35. **Разработване на графичния интерфейс със средствата на графична интегрирана среда за визуално моделиране на свойства и събития на компоненти**, многократно използване на потребителски дефинирани компоненти.
36. Въведение към **Java Applets**. **Основни методи** на *Applet*. Класът *JApplet* на *Swing* библиотеката и метода *paint*. **HTML тагове** за изобразяване на *Applet*. **Използване на параметри и четене на тези параметри** от *JApplet* приложения
37. Създаване на **многократно използвани графични компоненти** и вмъкването им в аплети и графични desktop приложения

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. 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 работа) по следната скала:

Софийски Университет “Климент Охридски”

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

Литература

- [1] **H. M. Deitel, P. J. Deitel, T. R. Nieto** “Java. How to Program”, 7th ed., Prentice Hall **2007**
ISBN 0-13-222220-5 (**основна**)
- [2] **Bruce Eckel** “Thinking in Java”, 2nd ed., Prentice Hall 2000
или българското й издание "Да мислим на Java" том 1 и 2, SoftPress, **2001**
- [3] **Adrian and Kathie Kingsley-Hughes**, “Beginning Programming“ **WROX 2005**

февруари, 2009