

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет автоматики и вычислительной техники

“УТВЕРЖДАЮ”

Декан АВТФ

профессор, д.т.н. Гужов
Владимир Иванович

“ ___ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Методы программирования и прикладные алгоритмы

ООП: специальность 090104.65 Комплексная защита объектов информатизации

Шифр по учебному плану: ОПД.Ф.3

Факультет: автоматики и вычислительной техники очная форма обучения

Курс: 2, семестр: 4

Лекции: 34

Практические работы: - Лабораторные работы: 34

Курсовой проект: - Курсовая работа: 4 РГЗ: -

Самостоятельная работа: 92

Экзамен: 4 Зачет: -

Всего: 160

Новосибирск

2011

Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования по направлению (специальности): 075400 Комплексная защита объектов информатизации.(№ 331 инф/сп от 14.04.2000)

ОПД.Ф.3, дисциплины федерального компонента

Рабочая программа обсуждена на заседании кафедры Вычислительной техники протокол № 1 от 28.01.2011

Программу разработал

доцент, к.т.н.

Васюткина Ирина Александровна

Заведующий кафедрой

профессор, д.т.н.

Губарев Василий Васильевич

Ответственный за основную образовательную программу

с.н.с., к.т.н.

Трушин Виктор Александрович

1. Внешние требования

Таблица 1.1

Шифр дисциплины	Содержание учебной дисциплины	Часы
ОПД.Ф.3	<p>ОПД.Ф.03 Методы программирования и прикладные алгоритмы</p> <p>современные технологии программирования; оценка качества программного обеспечения; общие принципы методы и средства проектирования архитектуры и структуры, проектирования логики, тестирования и отладки, документирования и сопровождения программного обеспечения с учетом повышенных требований к надежности программ и их защищенности от несанкционированного доступа; CASE-технологии, технологии виртуального программирования и объектно-ориентированного программирования; применение математических методов в проектировании надежного и защищенного программного обеспечения: функциональное программирование, логическое программирование; структуры данных и абстракции данных; оценка сложности алгоритмов; модели вычислений; алгоритмы сортировки, алгоритмы поиска; алгоритмы на графах; генерация случайных последовательностей; алгоритмы на подстановках; параллельные алгоритмы: методы проектирования параллельных алгоритмов, оценки сложности.</p> <p>Требования к профессиональной подготовленности выпускника по специальности 075400 - комплексная защита объектов информатизации</p> <p>Специалист по защите информации</p> <p>должен знать:</p> <ul style="list-style-type: none"> - основные положения теории информации, принципы построения систем обработки и передачи информации, основы семантического подхода к анализу информации-онных процессов; - современные аппаратные и программные средства вычислительной техники; - принципы организации информационных систем в соответствии с требованиями информационной 	160

	<p>защищенности, в том числе в соответствии с требованиями по защите государственной тайны;</p> <ul style="list-style-type: none"> - конструкцию и основные характеристики технических устройств хранения, обработки и передачи информации, потенциальные каналы утечки информации, характерные для этих устройств, способы их выявления и методы оценки опасности, основную номенклатуру и характеристики аппаратуры, используемой для перехвата и анализа сигналов в технических каналах утечки информации, методы и средства инженерно-технической защиты информации; - принципы и методы противодействия несанкционированному информационному воздействию на вычислительные системы и системы передачи информации; - основные правовые положения в области информационной безопасности и защиты информации. <p>должен владеть:</p> <ul style="list-style-type: none"> - технологией проектирования, построения и эксплуатации комплексных систем защиты информации; - методами научного исследования уязвимости и защищенности информационных процессов; - методиками проверки защищенности объектов информатизации на соответствие требованиям нормативных документов. <p>(Выдержка из ГОСВПО по специальности 090104 (075400) - "Комплексная защита объектов информатизации")</p>	
--	---	--

2. Особенности (принципы) построения дисциплины

Таблица 2.1

Особенности (принципы) построения дисциплины

Особенность (принцип)	Содержание
Основания для введения дисциплины в учебный план по направлению или специальности	Рабочая программа составлена на основании Государственного образовательного стандарта высшего профессионального образования (ГОСВПО) по направлению подготовки дипломированного специалиста 075400 - "Комплексная защита объектов информатизации" (регистрационный номер - 331 инф/сп, дата утверждения - 14

	<p>апреля 2000 г.).</p> <p>Индексы дисциплины в ГОСВПО ОПД.Ф.03 Цикл ОПД - Общие профессиональные дисциплины</p>
Адресат курса	Группы специальности 090104 (075400) - "Комплексная защита объектов информатизации"
Основная цель (цели) дисциплины	<p>Студенты дополнительно к объектно-ориентированному языку С++, изучаемому в дисциплине "Информатика", изучают объектно-ориентированный язык Java, проводят сравнение языков, изучают особенности разных объектно-ориентированных технологий.</p> <ul style="list-style-type: none"> - Дисциплина охватывает очень большой объем материала и может читаться с усилением акцентов на разных разделах, в зависимости от уровня подготовки студентов и их пожеланий. - Дисциплина рассчитана на студентов, владеющих приемами программирования и знающих язык объектно-ориентированного программирования С++. - Все разделы дисциплины изучаются на основе сравнения объектно-ориентированных технологий: С++ и Java. - Дисциплина имеет практическую часть - лабораторные работы (34 часа). На лабораторных работах студенты реализуют программы с использованием приемов объектно-ориентированного программирования на Java. - Курсовая работа выполняется по индивидуальному заданию. Примерные варианты заданий приведены в программе.
Ядро дисциплины	<p>Выдержка из ГОСТа: ОПД.Ф.03 Методы программирования и прикладные алгоритмы</p> <p>современные технологии программирования; оценка качества программного обеспечения; общие принципы методы и средства проектирования архитектуры и структуры, проектирования логики, тестирования и отладки, документирования и сопровождения программного обеспечения с учетом повышенных требований к надежности программ и их защищенности от несанкционированного доступа; CASE-технологии, технологии виртуального программирования и объектно-ориентированного программирования; применение математических методов в проектировании надежного и защищенного программного обеспечения: функциональное программирование, логическое программирование; структуры данных и абстракции данных; оценка сложности алгоритмов; модели вычислений; алгоритмы сортировки, алгоритмы поиска; алгоритмы на графах; генерация случайных последовательностей; алгоритмы на подстановках; параллельные алгоритмы: методы проектирования параллельных алгоритмов, оценки</p>

	сложности.
Связи с другими учебными дисциплинами основной образовательной программы	Данная дисциплина является логическим продолжением изучения объектно-ориентированных технологий, начало изучения которых было положено в дисциплине "Информатика", объектно-ориентированный язык С++
Требования к первоначальному уровню подготовки обучающихся	Знание языка программирования С++ и основ объектно-ориентированного программирования
Особенности организации учебного процесса по дисциплине	

3. Цели учебной дисциплины

Таблица 3.1

После изучения дисциплины студент будет

иметь представление	
1	разработки потоковых проложений, синхронизации потоков и передачи данных между потоками
3	о применении CASE- технологий в разработке программных систем
знать	
4	сравнительную характеристику наиболее популярных объектно-ориентированных языков программирования C++ и Java
5	принципы объектно-ориентированного программирования и их реализация в Java
6	какие программные средства используются для построения объектно-ориентированных программ
7	особенности информационных технологий, основанных на объектно-ориентированных языках (на примере Java)
8	основные определения элементов ООП (объект, класс, инкапсуляция, наследование, полиморфизм и т.д.)
9	средства и методы защиты объектно-ориентированных программ
уметь	
10	тестировать разработанное программное обеспечение
11	использовать пакеты стандартных классов для разработки оконных и многопоточковых приложений
12	реализовывать прикладные алгоритмы на Java
13	документировать разработанные программные продукты
14	использовать современные методы и среды создания программного обеспечения
иметь опыт (владеть)	
15	проектирования и реализации ОО программ
16	применения средств защиты программ в языке Java

4. Содержание и структура учебной дисциплины

Лекционные занятия

Таблица 4.1

(Модуль), дидактическая единица, тема	Часы	Ссылки на цели
Семестр: 4		
Модуль: Основные принципы ООП. Введение в язык Java.		
Дидактическая единица: основные понятия и модели, свойства объектно-ориентированных языков		
Java и World Wide Web. Развитие Java - технологий. Характеристики языка Java. Java и C++. Средства разработки Java приложений.	4	14, 4, 5, 6, 7

Объекты. Абстракция. Инкапсуляция. Интерфейсы. Сообщения. Модульность. Классификация. Наследование. Простейшие программы на Java. Основы Applet.		
Модуль: Основные элементы языка Java.		
Дидактическая единица: Типы данных: примитивные, ссылочные. Операции, операторы языка Java. Оценка сложности алгоритмов; модели вычислений; алгоритмы сортировки, алгоритмы поиска.		
Идентификаторы и стандарт Unicode. Комментарии. Типы данных. Простые типы данных. Ссылочные типы. Объявление переменных. Выражения и операции, приоритеты операций. Управляющие операторы. Массивы. Строки. Прикладные алгоритмы обработки массивов и строк.	6	12, 14, 8
Модуль: Основы Java: классы, наследование, пакеты.		
Дидактическая единица: системы объектов и классов; проектирование объектно-ориентированных программ: методы и алгоритмы. Интерфейсы.		
Описание класса. Создание объектов. Операция new. Методы и данные класса. Спецификаторы доступа. Многоуровневая иерархия классов. Реализация наследования. Переопределение методов. Полиморфизм. Абстрактные классы. Вложенные и внутренние классы. Пакеты.	6	10, 12, 14, 5
Модуль: Интерфейсы. Обработка исключений		
Дидактическая единица: Исключительные ситуации. Обработка исключений. Классы исключений.		
Определение и использование интерфейсов. Множественное наследование интерфейсов. Обработка исключений.	2	10, 12, 14, 16, 5, 8, 9
Модуль: АWT. Использование элементов графического интерфейса.		
Дидактическая единица: интерфейс: правила организации, методы и средства программирования. общие принципы методы и средства проектирования архитектуры и структуры, проектирования логики, тестирования и отладки ПО		
Классы АWT. Создание оконных приложений. Отображение информации в окне. Работа с графикой. Элементы управления. Менеджеры компоновки. Меню. Диалоговые окна. Работа с изображениями. Библиотека SWING. Классы JFC	3	10, 11, 14, 15, 16, 8, 9
Модуль: Обработка событий.		
Дидактическая единица: События. Модель делегирования событий. Слушатель события.		
Модели обработки событий. Классы событий.	2	10, 11, 12, 14,

Интерфейсы прослушивания событий. Использование модели делегирования событий. Классы - адаптеры. Анонимные внутренние классы.		15, 16, 8, 9
Модуль: Библиотека классов Java		
Дидактическая единица: Работа с унаследованным кодом. Библиотеки классов в Java.		
Основные классы - Object и Class. Классы - оболочки простых типов. Коллекции в Java. Сер-висные классы. RTTI. Организация вво-да/вывода через потоки. Сериализация.	3	10, 11, 12, 14, 15, 16, 8, 9
Модуль: Многопоточное программирование (многозадачность).		
Дидактическая единица: Многопоточность. Поток. Параллельные алгоритмы: методы проектирования параллельных алгоритмов, оценки сложности.		
Поточная модель Java. Создание потоков. Состояние потоков. Приоритеты. Синхронизация. Передача информации между потоками. Группы потоков. Потоки-демоны. Задачи параллельного программирования.	4	1, 11, 12, 14, 15, 16, 9
Модуль: Методы и средства проектирования архитектуры и структуры ПО.		
Дидактическая единица: современные технологии программирования; case-технологии		
Модели разработки ПО. Case-технологии. Тестирование и отладка. Критерии качества ПО. Документирование. Методы защиты ПО.	4	10, 13, 14, 15, 3, 6, 7

Лабораторная работа

Таблица 4.2

(Модуль), дидактическая единица, тема	Учебная деятельность	Часы	Ссылки на цели
Семестр: 4			
Модуль: Основные принципы ООП. Введение в язык Java.			
Дидактическая единица: основные понятия и модели, свойства объектно-ориентированных языков			
Основы программирования на Java. Разработка простого консольного приложения	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 12, 5, 6
Модуль: Основные элементы языка Java.			
Дидактическая единица: Типы данных: примитивные, ссылочные.			

Операции, операторы языка Java. Оценка сложности алгоритмов; модели вычислений; алгоритмы сортировки, алгоритмы поиска.			
Массивы, строки. Алгоритмы поиска и сортировки.	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 12, 5, 6
Модуль: Основы Java: классы, наследование, пакеты.			
Дидактическая единица: системы объектов и классов; проектирование объектно-ориентированных программ: методы и алгоритмы. Интерфейсы.			
Разработка классов в Java. Наследование.	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 12, 5, 8
Модуль: AWT. Использование элементов графического интерфейса.			
Дидактическая единица: интерфейс: правила организации, методы и средства программирования. общие принципы методы и средства проектирования архитектуры и структуры, проектирования логики, тестирования и отладки ПО			
Разработка оконного приложения	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 11, 12, 5, 6, 7, 8, 9
Модуль: Библиотека классов Java			
Дидактическая единица: Работа с унаследованным кодом. Библиотеки классов в Java.			
Потоки данных. Работа с локальными файлами.	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 12, 14, 16, 5, 9
Множественное наследование	Изучает тему, на	4	10, 12, 14,

интерфейсов. Сериализация.	готовых примерах программ. Выполняет индивидуальное задание на лаб. работу		15, 16, 5, 7, 8, 9
Коллекции	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	4	10, 11, 12, 14, 15, 16, 6, 7
Модуль: Многопоточное программирование (многозадачность).			
Дидактическая единица: Многопоточность. Поток. Параллельные алгоритмы: методы проектирования параллельных алгоритмов, оценки сложности.			
Многопоточковые приложения. Передача данных между потоками	Изучает тему, на готовых примерах программ. Выполняет индивидуальное задание на лаб. работу	6	1, 10, 11, 12, 14, 15, 5, 8, 9

5. Самостоятельная работа студентов

Семестр- 4, Курсовая работа

Примеры заданий:

Задание № 1. Создать игровую программу "Тетрис".

Задание № 2. Контроль успеваемости студентов. Разработать программу, реализующую учет баллов приема заданий по дисциплине. Страница содержит таблицу со списком студентов и количеством заданий по дисциплине. Студенты могут только посмотреть текущее состояние дел. Преподаватель, после ввода пароля, может проставлять баллы. Должна быть предусмотрена возможность сортировки списка студентов по алфавиту или по рейтингу. Предусмотреть возможность проставления экзаменационной оценки студентам или зачета. Преподаватель сам задает название предмета, количество заданий и экзамен/зачет, а также вводит фамилии студентов.

Задание № 3. Создать приложение, представляющую кафедру ВТ, в окне отображать несколько панелей, которые поочередно выходят на первый план. Первая панель - заглавная страница кафедры, следующая - рекламная информация о кафедре. Последующие панели содержат информацию о преподавателях кафедры, их фото и читаемые дисциплины. Программа работает в 2-х режимах: по щелчку мыши по кнопке далее или в автоматическом режиме.

Задание № 4. Рисование фракталов в Java.

Программа строит фракталы. Для получения изображения достаточно при помощи кнопок ">>" ("следующий") и "<<" ("предыдущий") выбрать фрактал (его название появляется строке состояния). Процесс рисования можно приостановить (кнопка "Остановить"), продолжить (кнопка "Продолжить") или начать заново (снова нажать на кнопку "Рисовать"). В программе заложено рисование 7 фракталов:

- o триадной кривой Коха
- o дракона Хартера-Хейтуэя
- o листа папоротника Барнсли
- o множества Жюлиа и 3 его видоизменений

Пользователю предоставляется возможность поэкспериментировать и получить совершенно отличные от начальных фракталы. Для этого достаточно изменять коэффициенты.

Семестр- 4, Подготовка к экзамену

Вопросы для подготовки к экзамену:

1. Основные принципы объектно-ориентированного программирования.
2. Технология Java. JVM. Сравнительная характеристика языков C++ и Java.
3. Именованные константы в Java. Примитивные типы данных. Ссылочные типы. Переменные.
4. Выражения и операции. Приведение типов. Управляющие операторы.
5. Массивы в Java. Создание и инициализация. Передача в функции. Клонирование массивов. Алгоритмы поиска заданного элемента.
6. Алгоритмы сортировки массивов.
7. Анализ сложности алгоритмов. Виды сложности. Классы скоростей роста сложности.
8. Объявление класса. Компоненты класса. Создание объектов класса. Модификаторы доступа к классам и членам классов.
9. Интерфейсы. Реализация интерфейсов. Множественное наследование интерфейсов.
10. Вложенные и локальные классы. Создание объектов этих классов.
11. Наследование. Реализация наследования. Переопределение методов родительского класса. Полиморфизм. Абстрактные классы.
12. Обработка исключений. Проверяемые и непроверяемые исключения. Создание пользовательских классов исключений.
13. Поточная модель Java. Базовые классы создания потоков. Состояние потоков. Приоритеты. Потоки-демоны.
14. Синхронизация потоков.
15. Передача информации между потоками.
16. Строки в Java. Обработка строк. Классы - оболочки простых типов.
17. Принципы построения коллекций в Java. Классы коллекций. Доступ к элементам коллекций.
18. Потоки ввода/вывода. Классификации потоков. Создание потоков ввода вывода. Буферизация.
19. Сериализация объектов. Управляемая сериализация.
20. События. Интерфейсы прослушивания событий. Модель делегирования событий.
21. События. Классы-адаптеры. Использование вложенных и анонимных внутренних классов для обработки событий.
22. Графические библиотеки Java. Принцип разработки графического интерфейса приложения. Формы. Диалоговые формы. Контейнеры.
23. Элементы управления GUI: менеджеры компоновки, меню, виды элементов управления.
24. Механизм RTTI. Классы Generic в Java.

Семестр- 4, Подготовка к занятиям

Подготовка к лекциям и лабораторным работам - 98 часов

Модуль 1. - 10 ч.

Основные принципы объектно-ориентированного программирования. Введение в язык Java.

Модуль 2. - 10 ч.

Основные элементы языка Java.

Модуль 3. - 10 ч.

Основы Java: классы, интерфейсы, пакеты.

Модуль 4 - 10 ч.

Наследование. Обработка исключений.

Модуль 5 - 10 ч.

AWT. Использование элементов графического интерфейса.

Модуль 6 - 10 ч.

Обработка событий.

Модуль 7 - 12 ч.

Многопоточное программирование (многопоточность).

Модуль 8 - 12 ч.

Библиотека Java. Символьные строки. Потоки ввода-вывода. Коллекции.

Модуль 9 - 14 ч.

Методы и средства проектирования архитектуры и структуры ПО.

6. Правила аттестации студентов по учебной дисциплине

Для аттестации студентов по дисциплине используется балльно-рейтинговая система. Рейтинг студента по дисциплине определяется как сумма баллов за работу в семестре (текущий рейтинг) и баллов, полученных в результате итоговой аттестации (экзамен).

В таблице приведено максимальное количество баллов, которое может набрать студент по видам учебной деятельности в течение семестра и диапазоны баллов, соответствующие минимальному и максимальному количеству баллов. Максимальная сумма баллов за семестр составляет 100 баллов (текущий рейтинг - 60 баллов, итоговая аттестация - 40 баллов).

Правила текущей аттестации:

1. В течение четвертого семестра необходимо представить и защитить 8 лабораторных работ и курсовую работу в сроки, установленные учебным графиком (см. таблицу).

2. К защите допускаются студенты, выполнившие лабораторные работы в полном объеме (все задания согласно варианту) и оформившие отчет по работе в соответствии с требованиями.

3. На защите предлагается два теоретических вопроса и один практический вопрос (по ходу выполнения работы).

4. Максимальное количество баллов 8 выставляется, если студент полностью ответил на все вопросы, без серьезных замечаний и недочетов.

5. Количество баллов 6-7 выставляется, если студент ответил на два вопроса из трех, причем один из вопросов - практический.

6. Минимальное количество баллов 4-5 выставляется, если студент ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.

7. При несвоевременной защите лабораторных работ оценка студента снижается на 10% , за каждый пропущенный срок защиты, согласно таблицы.

8. К защите курсовых работ допускаются студенты, выполнившие курсовую работу в полном объеме (все задания согласно варианту) и оформившие отчет по работе в соответствии с требованиями.

9. На защите предлагается два теоретических вопроса и один практический вопрос (по ходу выполнения работы).

10. Максимальное количество баллов 88-100 выставляется, если студент полностью выполнил задание на курсовую работу, отлично оформил пояснительную записку и ответил на все вопросы, без серьезных замечаний и недочетов.

11. Количество баллов 71-87 выставляется, если студент выполнил задание на курсовую работу, оформил пояснительную записку, но были допущены небольшие недочеты в работе или ее оформлении, ответил на два вопроса из трех, причем один из вопросов - практический.

12. Минимальное количество баллов 50-70 выставляется, если студент выполнил задание на курсовую работу, оформил пояснительную записку, но имеются существенные замечания по работе и ее оформлению, ответил на два вопроса из трех частично, с серьезными замечаниями, недочетами.

13. Передача лабораторной работы, курсовой работы назначается, если студент не ориентируется в учебном материале, не может объяснить ход и результаты выполнения работы. В случае передачи работы происходит потеря баллов (20% от максимального в зависимости от вида работы).

14. В случае представления и защиты курсовой работы с опозданием от учебного графика происходит потеря баллов (10 % за каждую неделю от максимально возможного).

Правила итоговой аттестации:

1. К экзамену допускаются студенты, сдавшие лабораторные работы и набравшие не менее 50% (30 баллов) по результатам текущего рейтинга.
2. Экзамен проводится в устной форме. В экзаменационный билет входят один теоретический вопрос и один практический (см. п.8).
3. Максимальное количество 36-40 баллов выставляется, если правильные ответы составляют 88% - 100%.
4. Количество баллов 29-35 выставляется, если правильные ответы составляют 73% - 87%.
5. Минимальное количество баллов 20-28 выставляется, если правильные ответы составляют 50% - 72%.

Таблица 1.

№п/п	Вид учебной работы (учебной деятельности)	Макс. кол-во баллов	Диапазоны
(неделя семестра)			
1.	Лабораторная работа №1 занятие по расписанию	4	2-4 Следующее
2.	Лабораторная работа №2 занятие по расписанию	8	4-8 Следующее
3.	Лабораторная работа №3 занятие по расписанию	8	4-8 Следующее
4.	Лабораторная работа №4 занятие по расписанию	8	4-8 Следующее
5.	Лабораторная работа №5 занятие по расписанию	8	4-8 Следующее
6.	Лабораторная работа №6 занятие по расписанию	8	4-8 Следующее
7.	Лабораторная работа №7 занятие по расписанию	8	4-8 Следующее
8.	Лабораторная работа №8	8	4-8 Зачетная неделя
	Итого по текущему рейтингу:	60	30-60
9.	Экзамен	40	20-40
	Итого за четвертый семестр:	100	50-100
10.	Курсовая работа	100	50-72 (удовл.) 73-87 (хор.) 88-100 (отл.)

17 неделя

7. Список литературы

7.1 Основная литература

В печатном виде

1. Васильев А. Н. Java. Объектно-ориентированное программирование : для магистров и бакалавров : базовый курс по объектно-ориентированному программированию : [учебное пособие] / А. Н. Васильев. - СПб. [и др.], 2011. - 395, [1] с. : ил., табл.
2. Эккель Б. Философия Java / Брюс Эккель ; [пер. с англ. Е. Матвеев]. - СПб. [и др.], 2010. - 637 с.
3. Соломон М. К. Oracle. Программирование на языке Java / Мартин К. Соломон, Нирва Мориссо-Леруа, Джули Басу ; [пер. И. Дранишников]. - М., 2010. - 484 с. : ил.
4. Машнин Т. С. Современные Java технологии на практике / Тимур Машнин. - СПб., 2010. - 552 с. : ил., табл. + 1 CD-ROM.
5. Рамбо Д. UML 2.0. Объектно-ориентированное моделирование и разработка / Дж. Рамбо, М. Блаха. - СПб. [и др.], 2007. - 540, [4] с. : ил.
6. Языки программирования [Электронный ресурс] : учебный комплекс INTUIT.ru / Интернет университет информационных технологий. - М., 2006. - 1 электрон. опт. диск (CD-ROM). - Загл. с экрана.
7. Крылов Е. В. Техника разработки программ. В 2 кн.. Кн. 1 : [учебник для вузов по направлениям "Информатика и вычислительная техника" и "Техника и технологии"] / Е. В. Крылов, В. А. Острейковский, Н. Г. Типикин. - М., 2007. - 374, [1] с. : ил. - Рекомендовано УМО.

В электронном виде

1. Языки программирования [Электронный ресурс] : учебный комплекс INTUIT.ru / Интернет университет информационных технологий. - М., 2006. - 1 электрон. опт. диск (CD-ROM). - Загл. с экрана.

7.2 Дополнительная литература

В печатном виде

1. Арнольд К. Язык программирования JAVA : [пер. с англ.] / Кен Арнольд, Джеймс Гослинг. - СПб, 1997. - 304 с.
2. Баженова И. Ю. Язык программирования Java. - М., 1997. - 288 с. : ил.
3. Нортон П. Программирование на Java. В 2 кн.. Кн. 1 : Пер. с англ.. - М., 1998. - 522 с. : ил.
4. Нортон П. Программирование на Java. В 2 кн.. Кн. 2 : Пер. с англ.. - М., 1998. - 383 с. : ил.. - В прилож.: CD-ROM.
5. Томас М. Секреты программирования для Internet на Java / М. Томас, П. Пател, А. Хадсон, Д. Болл. - СПб., 1997. - 640 с. : ил.
6. Мейнджер Д. JAVA: основы программирования : Пер. с англ.. - Киев, 1997. - 320 с.
7. Нотон П. Полный справочник по JAVA : Пер. с англ.. - Киев, 1997. - 592с. : ил.
8. Джамса К. Java сегодня : пер. с англ. / Крис Джамса. - Минск, 1996. - 412 с. : ил.
9. Пушкарева Г. В. CASE-технологии: практическая работа в Rational Rose : учебное пособие / Г. В. Пушкарева ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 58, [1] с. : ил.

В электронном виде

1. Block M. Java-Intensivkurs [electronic resource] : : In 14 Tagen lernen Projekte erfolgreich zu realisieren // von Marco Block. - Berlin, Heidelberg ;, 2010. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-3-642-03955-3>

2. Rischpater R. Beginning Java™ ME Platform [electronic resource] // by Ray Rischpater. - Berkeley, CA :, 2009. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-4302-1062-7>
3. Nielsen F. A Concise and Practical Introduction to Programming Algorithms in Java [electronic resource] // by Frank Nielsen. - London :, 2009. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-84882-339-6>
4. Solymosi A. Grundkurs Algorithmen und Datenstrukturen in JAVA [electronic resource] : : Eine Einführung in die praktische Informatik // by Andreas Solymosi, Ulrich Grude. - Wiesbaden :, 2008. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-3-8348-9522-6>
5. Silberbauer C. Einstieg in Java und OOP [electronic resource] // von Christian Silberbauer. - Berlin, Heidelberg :, 2008. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-3-540-78618-4>
6. Klawonn F. Introduction to Computer Graphics [electronic resource] : : Using Java 2D and 3D // by Frank Klawonn. - London :, 2008. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-84628-848-7>
7. Kayal D. Pro Java™ EE Spring Patterns [electronic resource] : : Best Practices and Design Strategies Implementing Java™ EE Patterns with the Spring Framework // by Dhrubojyoti Kayal. - Berkeley, CA :, 2008. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-4302-1010-8>
8. Friesen J. Beginning Java™ SE 6 Platform [electronic resource] : : From Novice to Professional / by Jeff Friesen. - Berkeley, CA :, 2007. : v.: digital // Springer eBooks. - Режим доступа: <http://dx.doi.org/10.1007/978-1-4302-0246-2>
9. Пушкарева Г. В. CASE-технологии: практическая работа в Rational Rose : учебное пособие / Г. В. Пушкарева ; Новосиб. гос. техн. ун-т. - Новосибирск, 2006. - 58, [1] с. : ил.. - Режим доступа: <http://www.library.nstu.ru/fulltext/metodics/2006/pushk.rar>

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

В печатном виде

1. Вебер Д. Технология Java в подлиннике : Учебник. Практическое пособие. Справочник : наиболее полное руководство : для прогаммистов] / Д. Вебер ; [пер. с англ.: Сергей Иноземцев и др.]. - СПб., 1997. - XXIV, 1079 с. : ил. + CD-ROM.
2. Федотова Д. Э. CASE-технологии : [практикум] / Д. Э. Федотова, Ю. Д. Семенов, К. Н. Чижик. - М., 2005. - 157 с. : ил.

8.2 Программное обеспечение

1. Oracle, JDK 7, Инструмент разработки на языке Java
2. Oracle, Net Beans 7.0.1, Инструментальная среда разработки

9. Контролирующие материалы для аттестации студентов по дисциплине

Темы экзаменационных задач.

1. Обработка массивов данных.
2. Разработать приложение с GUI. Использовать менеджеры компоновки, элементы управления: кнопки, текстовые поля, радиокнопки, чекбоксы, списки.
3. Работа с потоками ввода/вывода.
4. Разработать иерархию классов.
5. Создание многопоточной программы.

Примеры задач.

Задача

1. Создать программу, в которой строки текста, вводимые с клавиатуры записываются в файл. Использовать буферизированный вывод данных.

Задача

2. Создать приложение, использующее два потока (thread). Один из потоков генерирует данные (случайные числа), а второй их считывает. Синхронизировать работу потоков.

Задача

3. Разработать иерархию классов: печатное издание (абстрактный класс), книга, учебник. Используя класс-коллекцию LinkedList, создать список объектов класса. Иметь возможность сохранять в файл этот список и восстанавливать, используя механизм сериализации.