

ФЕДЕРАЛЬНОЕ АГЕНСТВО ПО ОБРАЗОВАНИЮ РОССИИ
ГОУ ВПО «ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Физико-математический факультет
Кафедра информатики

УЧЕБНАЯ ПРОГРАММА

по дисциплине

«Системное и прикладное программное обеспечение»

Специальность: 010501. Прикладная математика и информатика

Квалификация: Математик, системный программист

Составитель:

к.ф.-м.н., доц. Никольский Д. Н.

- 2 -

2008 г.

ВЫПИСКА ИЗ ПРОТОКОЛА
ЗАСЕДАНИЯ КАФЕДРЫ ИНФОРМАТИКИ
ОБ УТВЕРЖДЕНИИ ПРОГРАММЫ КУРСА
СИСТЕМНОЕ И ПРИКЛАДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Данная программа составлена в соответствии с государственным образовательным стандартом высшего профессионального образования по специальности «010501. Прикладная математика и информатика», квалификации «Математик, системный программист».

Номер государственной регистрации
199 ЕН/СП

Выписка верна.

Протокол № 1 от 4 сентября 2007 года

Зав. Кафедрой

Никольский Д.Н.

Требования к обязательному минимуму содержания основной образовательной программы подготовки математика, системного программиста по специальности 010501 — Прикладная математика и информатика

Общепрофессиональные дисциплины

ОПД.Ф.06 Системное и прикладное программное обеспечение

Содержание государственного образовательного стандарта:

Основные этапы, методы, средства и стандарты разработки программного обеспечения; системы программирования (принципы организации, состав и схема работы); основные типы операционных систем, принципы управления ресурсами в операционной системе; сети ЭВМ и протоколы передачи информации.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

по дисциплине «Системное и прикладное программное обеспечение»

Данный курс предназначен для студентов физико-математического факультета специальности 010501 Прикладная математика и информатика.

Цели и задачи дисциплины: общая и специальная профессиональная подготовка, развитие и углубление знаний и навыков в области компьютерных технологий.

Данный курс является логическим продолжением дисциплины «Информатика» 1-го курса обучения, посвященной программированию на языке C++. В ходе изучения этого курса студенты совершенствуют навыки программирования на C++ путем изучения библиотек STL (Стандартной библиотеки шаблонов) и Qt (Библиотеки для разработки GUI-приложений) посредством чего глубоко знакомятся с этапами, методами, средствами и стандартами разработки программного обеспечения, изучают принципы организации, состав и схемы работы систем программирования, а также знакомятся с принципами управления ресурсами в операционной системе, сетями ЭВМ, протоколами передачи информации и так далее.

При изучении библиотеки STL студенты знакомятся со способами организации динамических структур данных и алгоритмы их обработки. При изучении библиотеки Qt студенты знакомятся с организацией приложений с графическим интерфейсом, а затем на примере кроссплатформенных классов Qt изучают элементы системного программирования (процессы, потоки, взаимодействие с базами данных, клиент-серверное приложение, событийный механизм, файловая система, взаимодействие между процессами).

Основные знания, умения, навыки:

Студенты должны знать:

- динамические структуры данных; основные алгоритмы
- основные контейнеры STL, основные алгоритмы STL; алгоритмы и динные Boost C++ Libraries;
- общие принципы организации GUI-приложений; понятие иерархии графических элементов; виджет верхнего уровня; назначение менеджеров и классов размещения;
- механизм сигналов и слотов;
- элементы GUI-приложения и их реализацию в библиотеке Qt;
- событийный механизм и способы перехвата событий;
- организацию двумерной графики; исполнитель команд рисования, его инструменты, контекст рисования;
- организация 3-х мерной графики, пример классов в Qt, элементы OpenGL;
- механизм многопоточных приложений, организация процессов, примеры классов в Qt;
- взаимодействие с базами данных, примеры классов в Qt;
- сетевые протоколы, протокол TCP/IP, высокоуровневые протоколы;
- особенности организации клиент-серверных приложений и классы Qt, их реализующий;
- механизм взаимодействия между процессами.
- теоретические основы современных ОС
- особенности построения, классификацию и области применения ОС
- основные понятия стандарта POSIX
- принципы построения параллельных вычислительных систем и сетей
- современные подходы к проектированию сложного системного программного и основы соответствующих инструментальных средств

Математик, системный программист должен уметь:

- применять алгоритмы стандартной библиотеки шаблонов (STL) для решения практических задач

- применять динамические структуры стандартной библиотеки шаблонов (STL) для решения практических задач

- разрабатывать простейшие приложения с графическим интерфейсом

- разрабатывать однодокументные приложения

- разрабатывать многодокументные приложения

- уметь разрабатывать приложения, использующие механизм обработки событий

- разрабатывать приложения с 2-х и 3-х мерной графикой

- организовывать многопоточные приложения и запускать процессы

- разрабатывать графический интерфейс к БД

- организовывать клиент-серверные приложения

- проектировать и создавать отдельные компоненты операционных систем

- управлять ОС при помощи стандартных для неё программных интерфейсов

- создавать приложения для работы с программными интерфейсами конкретных

ОС и кроссплатформенные

Математик, системный программист должен обладать навыками: применения современных технологий разработки программного обеспечения для решения прикладных задач из областей науки, техники, экономики и управления; самостоятельной и коллективной разработки программных комплексов различной степени сложности.

Полная трудоемкость курса 102 часа, из них 72 аудиторных занятий, которые проводятся в форме лекций , и 30 часа – самостоятельной работы. В третьем семестре проводится 36 лекционных занятий и предполагается 16 часов самостоятельной работы, в четвертом семестре проводится 36 часов в форме лекций и 14 часов в форме самостоятельной работы.

Форма контроля: зачет в третьем семестре, экзамен – в четвертом.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Содержание курса	Количество часов				
		Всего часов	Лекции	Лабор. занятия	Сам. занятия	Литература
	3 семестр					
1.	РАЗДЕЛ I. Разработка программного обеспечения.	4	2	0	2	
2.	РАЗДЕЛ II. Стандартная библиотека шаблонов.	6	4	0	2	
3.	РАЗДЕЛ III. Разработка приложений с графическим интерфейсом.	16	10	0	6	
4.	РАЗДЕЛ IV. Графика	4	2	0	2	
5.	РАЗДЕЛ V. Системное программирование	20	14	0	6	
	Итого по 3-му семестру	50	32	0	18	
	4 семестр					
1.	РАЗДЕЛ VI Операционные системы	50	36	0	14	

Лабораторные занятия выполняются в рамках дисциплины «Практикум на ЭВМ»

СОДЕРЖАНИЕ ПРОГРАММЫ ДИСЦИПЛИНЫ «Системное и прикладное программное обеспечение»

3-й семестр

РАЗДЕЛ I. Разработка программного обеспечения.

1. Этапа процесса разработки ПО, циклы разработки ПО, проектирование.
2. Структурный и объектно-ориентированные подходы к разработке ПО.
3. Методы разработки алгоритмов.
4. Системы программирования. Обзор в основных ОС.

РАЗДЕЛ II. Стандартная библиотека шаблонов.

1. Проектирование контейнеров. Итераторы. Обзор STL.
2. Организация библиотеки STL. Основные требования, предъявляемые к STL. Общие методы и операции контейнеров.
3. Последовательные контейнеры. Контейнерные адаптеры.
4. Ассоциативные контейнеры. Отображение, мультиотображение, множество и мультимножество.
5. Алгоритмы и объекты функций.

РАЗДЕЛ III. Разработка приложений с графическим интерфейсом.

1. Общие принципы организации приложений с графическим интерфейсом. Особенности в Qt.
2. Виджеты общего назначения. Менеджеры и классы размещения.
3. Разработка одно- и много- документных приложений

РАЗДЕЛ IV. Графика

1. Организация двумерной графики.
2. Трехмерная графика. Элементы OpenGL.

РАЗДЕЛ V. Системное программирование

1. Работа с файлами
2. Событийный механизм.
3. Интерфейс к БД
4. Многопоточные приложения. Процессы.
5. Протокол ТСР/ІР. Клиент-серверные приложения. Высокоуровневые протоколы.
6. Межпроцессное взаимодействие.

4-й семестр

РАЗДЕЛ VI Операционные системы

1. Ведение
2. Процессы и потоки

3. Управление памятью
4. Архитектура процессоров x86 для организации мультипрограммных операционных систем
5. Управление вводом-выводом
6. Файловые системы
7. Организация параллельных взаимодействующих вычислений
8. Проблема тупиков
9. Архитектура операционных систем
10. Обзор современных операционных систем

ПЛАНЫ ЛАБОРАТОРНЫХ ЗАНЯТИЙ ПО ДИСЦИПЛИНЕ

3-й семестр

1. Проектирование контейнеров. Итераторы [1-4].
2. Вектор[1-4].
3. Список[1-4].
4. Ассоциативные контейнеры[1-4].
5. Адаптеры[1-4].
6. Алгоритмы. Объекты функций. Функциональная композиция[1-4].
7. Сборка приложений из стандартных виджетов [5].
8. Создание собственного класса для виджета верхнего уровня [5].
9. Одно-документное приложение [5].
10. Много-документное приложение [5].
11. Графика [5].
12. Обработка событий [5].
13. Таймер. Игровое приложение.[5].
14. Работа с файловой системой [5].
15. Многопоточное приложение [5].
16. Интерфейс к базам данных [5].
17. Разработка клиент-серверного приложения [5].
18. Комплексное задание [1-5].

4-й семестр

1. Вводное занятие.
2. Обработка сложных динамических структур данных (ч. 1)
3. Обработка сложных динамических структур данных (ч. 2)
4. Обработка сложных динамических структур данных (ч. 3)
5. Обработка сложных динамических структур данных (ч. 4)
6. Обработка сложных динамических структур данных (ч. 5)
7. Работа с файлами при помощи «локаторов».
8. Модель процесса. Дисциплины планирования.
9. Моделирование потоков уровня пользователя (ч. 1).
10. Моделирование потоков уровня пользователя (ч. 2).
11. Управление памятью (ч. 1).
12. Управление памятью (ч. 2).
13. Моделирование файловой системы (ч. 1).
14. Моделирование файловой системы (ч. 2).
15. Проблемы межпроцессного взаимодействия (ч. 1).
16. Проблемы межпроцессного взаимодействия (ч. 2).
17. Архитектурные особенности ОС.
18. Зачётная лабораторная работа (комплексное задание).

Содержание и виды самостоятельной работы.

Самостоятельная работа студента заключается в проработке лекций и дополнительной литературы (в т. ч. рекомендованной преподавателем), подготовке к лабораторным работам (используя лекции и учебные пособия) и подготовке к зачету. Объем самостоятельной работы примерно равен объему аудиторных занятий.

СПИСОК ОСНОВНОЙ ЛИТЕРАТУРЫ

1. Страуструп Б. Язык программирования С++. М.: Бином, 2001. 1099 с.
2. Джосьютис Н. С++ Стандартная библиотека для профессионалов. СПб Питер, 2004. 730 с.
3. Липпман С.Б. Основы программирования на С++. М.: Вильямс, 2002. 256 с.
4. Подбельский В.В. Язык С++. Учебное пособие. М.: Финансы и статистика, 1995. 560 с.
5. Шлее М. Qt. Профессиональное программирование на С++. СПб.: БХВ-Петербург, 2005. 544 с.
6. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. — СПб.: Невский диалект, 2001.
7. Таненбаум Э. Современные операционные системы. – СПб.: Питер, 2002.
8. Гордеев А. В. Операционные системы: Учебник для вузов. – СПб.: Питер, 2004.
9. Цилькер Б. Я., Орлов С. А. Организация ЭВМ и систем. Учебник для вузов. – СПб.: Питер, 2004
10. Gutknecht J., Wirth N. Project Oberon - The Design of an Operating System and Compiler.
– <http://www.oberon.ethz.ch/>

СПИСОК ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ

1. Working draft, Standart for Programming Language C++. (электронный)
2. Standart Template Library Programmer's Guide (электронная, документация к компилятору)
3. Arthur Griffith. KDE 2/ Qt programming bible. IDG Books Worldwide, Inc, 722 p. (электронная)
4. Qt Reference Documentation (Open Source Edition)
5. Лорин Г., Дейтел Х. М. Операционные системы: Пер. с англ. — М.: Финансы и статистика, 1984.
6. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация: 3-е издание — СПб.: Питер, 2007.
7. Орлов С. А. технологии разработки программного обеспечения: Учебник для вузов. — СПб.: Питер, 2004.
8. Стивенс У. UNIX: взаимодействие процессов. — СПб.: Питер, 2003.
9. Muller P. The Active Object System – Design and Multiprocessor Implementation. – PhD Dissertation, ETH Zurich, 2002.
10. Свердлов С. З. Языки программирования и методы трансляции: Учебное пособие. — СПб.: Питер, 2007.