

Министерство образования и науки
САРАТОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО

Кафедра дискретной математики и информационных технологий

РАБОЧАЯ ПРОГРАММА

по дисциплине	«Технологии программирования»
для специальности	230101 — « Вычислительные машины, комплексы, системы и сети»
реализуемой на факультете компьютерных наук и информационных технологий	

Саратов, 2011 год

Рабочая программа составлена в соответствии с Государственным образовательным стандартом высшего профессионального образования по специальности 230101 – «Вычислительные машины, комплексы, системы и сети» (номер государственной регистрации 224 тех/дс. от «27» марта 2000 г.)

ОДОБРЕНО:

Председатель научно-методической комиссии факультета КНиИТ

Салий В.Н.

« _____ » _____ 2011 г.

УТВЕРЖДАЮ:

Проректор по учебно-методической работе

Елина Е.Г.

« _____ » _____ 2011 г.

СОГЛАСОВАНО:

Декан факультета КНиИТ

Федорова А.Г.

« _____ » _____ 2011 г.

Вид учебной работы	Бюджет времени по формам обучения, час.		
	очная	очно-заочная	заочная
Аудиторные занятия, всего	72		18
В том числе:			
- лекции;	36		9
- лабораторные (практические)	36		9
- семинарские	□		-
Самостоятельная работа студентов	94		148
Зачёты (№ семестра)	3		-
Экзамены (№ семестра)	3		3
Контрольные работы (кол-во/семестр)	-		-
Курсовая работа	□		-

Заведующий кафедрой ДМиИТ

Тяпаев Л.Б.

Автор

Синельников Е.А.

« _____ » _____ 2011 г.

Раздел 1. Организационно-методическое сопровождение

Курс «Технологии Программирования» для студентов факультета компьютерных наук и информационных технологий обеспечивается кафедрой дискретной математики и информационных технологий. Курс читается в третьем семестре.

Цель дисциплины – дать студентам систематизированные представления о принципах построения и проектирования программных систем, а также сформировать практические навыки реализации формализованного описания процессов обработки данных.

1. В результате изучения курса студент должен знать:
 - принципы проектирования программных систем;
 - организацию процесса проектирования программного обеспечения;
 - методологию объектно-ориентированного программирования;
 - технологические средства разработки программного обеспечения;
 - методы отладки и тестирования программ;
 - методы защиты программ и данных.
2. В результате изучения курса студент должен уметь:
 - использовать методы декомпозиции и абстракции при проектировании ПО;
 - применять средства разработки программного обеспечения: инструментальные среды разработки, средства поддержки проекта, отладчики;
 - документировать и оценивать качество программных продуктов;
 - проектировать пользовательские интерфейсы.
3. Дисциплина связана с дисциплинами:

Информатика (Информационные технологии. Технические и программные средства информационных технологий. Основные виды обработки данных).

Программирование на языке высокого уровня (Основы алгоритмизации и программирования. Структуры данных. Способы конструирования и верификации программ).

Раздел 2. Тематический план учебной дисциплины

Наименование раздела, подраздела, темы лекции	Бюджет учебного времени					Форма текущего и итогового контроля
	Всего	в том числе				
		Лекции	Лабораторные и практические занятия	Семинарские занятия	Самостоятельная работа	
Очная полная программа						
1. Проблемы разработки сложных программных систем. Исторический и социальный контекст программирования.	6	4			2	
2. Объектно-ориентированный подход к проектированию программных систем	20	4	6		10	
3. Жизненный цикл и процессы разработки ПО	18	4	4		10	
4. Унифицированный процесс разработки и экстремальное программирование	12	2			10	
5. Анализ предметной области и требования к ПО	10	2			8	
6. Качество ПО и методы его контроля	16	4	4		8	
7. Архитектура программного обеспечения	20	4	6		10	
8. Образцы проектирования	16	4	2		10	
9. Принципы создания пользовательского интерфейса	16	2	4		10	
10. Компонентные технологии и разработка распределённого ПО	16	2	4		10	
11. Управление разработкой ПО	16	4	6		6	
ИТОГО:	166	36	36		94	Экзамен и отчёт по практике
Заочная полная программа						
1. Проблемы разработки сложных программных систем. Исторический и социальный контекст программирования.	6	1			5	
2. Объектно-ориентированный подход к проектированию программных систем	20	1	2		17	
3. Жизненный цикл и процессы разработки ПО	18	1	1		16	
4. Унифицированный процесс разработки и экстремальное программирование	12				12	
5. Анализ предметной области и требования к ПО	10				10	
6. Качество ПО и методы его контроля	16	1	1		14	
7. Архитектура программного обеспечения	20	1	1		18	
8. Образцы проектирования	16	1	1		14	
9. Принципы создания пользовательского интерфейса	16	1	1		14	

10. Компонентные технологии и разработка распределённого ПО	16	1	1		14	
11. Управление разработкой ПО	16	1	1		14	
ИТОГО:	166	9	9		148	Экзамен и отчёт по практике

Раздел 3. Содержание учебной дисциплины

Тема 1. Проблемы разработки сложных программных систем

Понятие сложной программы и отличия сложных программ от простых. Основные проблемы разработки сложных программ. Понятие информационной среды процесса обработки данных. Программа как формализованное описание процесса. Понятие о программном средстве. Понятие ошибки в программном средстве. Не конструктивность понятия правильной программы. Надёжность программного средства. Технология программирования как технология разработки надёжных программных средств. Роль в обществе компьютеров и программирования, информатизация общества. Применение инструментальных средств разработки ПО.

Тема 2. Объектно-ориентированный подход к проектированию программных систем

Введение в объектно-ориентированный подход к разработке и реализации прикладных программных систем. Методология объектно-ориентированного программирования. Языки программирования, предназначенные для реализации программ в рамках объектно-ориентированного подхода. Специфики процедур и данных. Статические и глобальные объявления. Особенности реализации объектно-ориентированного подхода не объектно-ориентированных языках программирования – сущность возможностей предоставляемых объектно-ориентированным подходом.

Тема 3. Жизненный цикл и процессы разработки ПО

Специфика разработки программных средств. Жизненный цикл ПО, виды деятельности, роли заинтересованных лиц, процессы жизненного цикла, процесс разработки ПО, стандарты жизненного цикла ПО, модель зрелости возможностей модели жизненного цикла ПО, каскадная модель жизненного цикла, итеративная модель жизненного цикла, спиральная модель жизненного цикла. Набор стандартов, регулирующих процессы разработки ПО в целом.

Тема 4. Унифицированный процесс разработки и экстремальное программирование

«Тяжёлые» процессы разработки, «живые» методы разработки, унифицированный процесс разработки Rational (RUP), экстремальное программирование (XP), модели ПО.

Тема 5. Анализ предметной области и требования к ПО

Анализ предметной области, схема Захмана, модели предметной области, диаграммы потоков данных, диаграммы сущностей и связей, функции ПО, требования к ПО, варианты использования, действующие лица, диаграммы вариантов использования.

Тема 6. Качество ПО и методы его контроля

Качество ПО, функциональность, надёжность, удобство использования, производительность, удобство сопровождения, переносимость, методы контроля качества, тестирование, проверка свойств ПО на моделях, ошибки в ПО. Методы отладки и тестирования программ. Документирование и оценка качества программных продуктов. Методы защиты программ и данных.

Тема 7. Архитектура программного обеспечения

Задачи проектирования программных систем. Использование декомпозиции и абстракции при проектировании ПО. Архитектура ПО, компонент архитектуры, представление архитектуры, сценарий использования, методы оценки архитектуры, диаграммы классов, диаграммы взаимодействия, диаграммы сценариев, диаграммы компонентов, диаграммы развёртывания. Методы проектирования структуры ПО.

Тема 8. Образцы проектирования

Образец проектирования, классификация образцов проектирования. Примеры образцов анализа и архитектурных стилей: образец анализа, образец организации, образец проектирования «подписчик», образец процесса, архитектурный стиль «каналы и фильтры», архитектурный стиль «многоуровневая система». архитектурный стиль «данные–представление–обработка», идиома «шаблонный метод», инспекция программ по Фагану.

Тема 9. Принципы создания пользовательского интерфейса

Основные факторы удобства использования ПО, психофизиологические особенности человека, методика проектирования, ориентированного на удобство использования, модель ролей пользователей, модель задач, модель содержимого интерфейса, эвристическое инспектирование интерфейса, тестирование удобства использования. Многооконные интерфейсы и проектирование интерфейсов с пользователем. Структуры диалогов в интерфейсах с пользователем. Организации поддержки пользователей. Примеры реализации интерфейсов с пользователем с использованием графических библиотек и пакетов.

Тема 10. Компонентные технологии и разработка распределённого ПО

Программный компонент, интерфейс, программный контракт, компонентная модель, компонентная среда, базовые службы компонентной среды, распределённое ПО, прозрачность, открытость, масштабируемость, безопасность, синхронное и асинхронное взаимодействие, удалённый вызов процедур, транзакция.

Тема 11. Управление разработкой ПО

Организация процесса проектирования программного обеспечения. Основные виды деятельности, входящие в компетенцию руководителей проектов, аспекты управления ресурсами, персоналом, рисками и коммуникациями проекта. особенности управления проектами по созданию ПО. Средства поддержки управления проектами.

Раздел 4. Перечень основной и дополнительной литературы

Основная литература

1. Кулямин В. В. Технологии программирования. Компонентный подход. — М.: Интернет-Ун-т Информ. Технологий, 2007 ; М.: БИНОМ. Лаб. знаний, 2007.
2. Синельников Е. А. Курс. Технологии программирования. — 2010.)
<http://course.sgu.ru/course/view.php?id=16> (доступен гостевой вход)

Дополнительная литература

1. Терехов А. Н. Технологии программирования: учебное пособие — М.: Интернет-Ун-т Информ. Технологий, 2007 ; М.: БИНОМ. Лаб. знаний, 2007.
2. Лунаев В. В. Программная инженерия. Методологические основы. — М.: ТЕИС, 2006.
3. Вязовик Н. А. Программирование на Java. — М.: Интернет-Университет Информ. Технологий, 2003.
<http://www.intuit.ru/department/pl/javapl/>
4. Мацяшек Л.А., Лионг Б.Л. Практическая программная инженерия на основе учебного примера. — М.: БИНОМ. Лаб. знаний, 2009.
5. Жоголев Е. А. Технология программирования. — М.: Науч. Мир, 2004.
6. Непейвода Н. Н. Основания программирования. — Ижевск: Изд. Ин-та компьютер. исслед., 2003.
7. Леоненков А. В. Самоучитель UML. — СПб.: БХВ-Петербург, 2004.
8. К. Бек. Экстремальное программирование: разработка через тестирование .. — СПб.: Питер, 2003.
9. Кулямин В. В. Технологии программирования. Компонентный подход. — 2006.
<http://panda.ispras.ru/~kuliamin/sdt-course.html>
<http://www.intuit.ru/department/se/compprog/>
10. J. Gosling, B. Joy, G. Steele, and G. Bracha. Java Language Specification, 3-rd edition. Addison Wesley Professional, 2005.
<http://people.toiit.sgu.ru/Sinelnikov/PT/Java/langspec-3.0.pdf>
<http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf>
11. C# Language Specification. Standard ECMA-334. 4-th edition, June 2006.
<http://people.toiit.sgu.ru/Sinelnikov/PT/dotNet/Ecma-334.pdf>
<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>
12. Жоголев Е. А. Введение в технологию программирования (конспект лекций). — М.: ДИАЛОГ-МГУ, 2004.
<http://lib.mexmat.ru/books/69295>
13. Леоненков А. В. Нотация и семантика языка UML. — 2005.
<http://www.intuit.ru/department/pl/umlbasics/>
14. Мейер Б. Основы объектно-ориентированного программирования. — 2005.
<http://www.intuit.ru/department/se/ooobases/>
15. Guide to the Software Engineering Body of Knowledge — 2004.
<http://www.computer.org/portal/web/swebok>
16. G. Kiefer. Extreme Programming Considered Harmful for Reliable Software Development. AVOCA Technical Report, 2002.
<http://people.toiit.sgu.ru/Sinelnikov/PT/Kiefer/ExtermeProgramming.pdf>
17. Kroll, The Spirit of the RUP. — 2001.
<http://people.toiit.sgu.ru/Sinelnikov/PT/IBM/developerworks/rational/TheSpiritoftheRUPDec01.pdf>
<http://www.ibm.com/developerworks/rational/library/content/RationalEdge/dec01/TheSpiritoftheRUPDec01.pdf>

18. *M. C. Paulk, B. Curtis, M. B. Chrissis, and C. V. Weber.* Capability Maturity Model for Software, Version 1.1, SEI Technical Report CMU/SEI-93-TR-024, Software Engineering Institute, Pittsburgh, Feb. 1993.
<http://people.toiit.sgu.ru/Sinelnikov/PT/SEI/reports/93tr024.pdf>
<http://www.sei.cmu.edu/reports/93tr024.pdf>
19. *M. C. Paulk, C. V. Weber, S. M. Garcia, M. B. Chrissis, and M. Bush.* Key Practices of the Capability Maturity Model, Version 1.1, SEI Technical Report CMU/SEI-93-TR-025, Software Engineering Institute, Pittsburgh, Feb. 1993.
<http://people.toiit.sgu.ru/Sinelnikov/PT/SEI/reports/93tr025.pdf>
<http://www.sei.cmu.edu/reports/93tr025.pdf>
20. Capability Maturity Model Integration (CMMI), Version 1.1. CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPP/SS, V1.1). Continuous Representation. SEI Technical Report CMU/SEI-2002-TR-011, Software Engineering Institute, Pittsburgh, March 2002.
<http://people.toiit.sgu.ru/Sinelnikov/PT/SEI/reports/02tr011.pdf>
<http://www.sei.cmu.edu/reports/02tr011.pdf>
21. Capability Maturity Model Integration (CMMI), Version 1.1. CMMI for Systems Engineering, Software Engineering, Integrated Product and Process Development, and Supplier Sourcing (CMMI-SE/SW/IPP/SS, V1.1). Staged Representation. SEI Technical Report CMU/SEI-2002-TR-012, Software Engineering Institute, Pittsburgh, March 2002.
<http://people.toiit.sgu.ru/Sinelnikov/PT/SEI/reports/02tr012.pdf>
<http://www.sei.cmu.edu/reports/02tr012.pdf>

Раздел 5. Перечень средств обучения.

Лабораторные занятия проводятся в дисплейном классе с выходом в Internet.

Необходимое программное обеспечение:

1. ОС Linux или ОС Windows;
2. консольный файловый менеджер (Far под Windows или mc под Linux)
3. компилятор C++ (GNU C++ Compiler или Microsoft Visual C++)
4. комплект средств разработчика для Java — Java Development Kit (JDK)
5. среда разработки (Qt Creator, NetBeans, Eclipse, Microsoft Visual Studio 2008);
6. система контроля версий Git (опционально средства интеграции со средами разработки);
7. браузер Mozilla Firefox (версии не ниже 3.6.13) с установленным плагином FireBug;
8. браузер Google Chromium;
9. комплект из лабораторных работ по темам:

Тема 1. Объектно-ориентированный подход к проектированию программных систем

Тема 2. Управление разработкой ПО

Тема 3. Жизненный цикл и процессы разработки ПО

Тема 4. Образцы проектирования

Тема 5. Архитектура программного обеспечения

Тема 6. Компонентные технологии и разработка распределённого ПО

Раздел 6. Темы вопросов к курсу в экзаменационных билетах.

1. Сложные и «большие» программные продукты. Отличительные особенности и основные свойства сложных программных продуктов.
2. Программная инженерия. Организационные, инженерные и технические аспекты разработки ПО
3. Модули и подсистемы. Понятие интерфейса модуля и подсистемы. Внутреннее и внешнее окружения.
4. Контекст времени исполнения при взаимодействии модулей и подсистем. Предусловия и постусловия в интерфейсах.
5. Принципы работы со сложными системами: абстракция, модульность, переиспользование. Разбиение системы на модули.
6. Принципы работы со сложными системами: адекватность, полнота, ортогональность и простота интерфейсов, разделение ответственности.
7. Жизненный цикл ПО: виды деятельности, артефакты и роли.
8. Модели жизненных циклов ПО: каскадная, итеративная и спиральная модели
9. Унифицированный процесс разработки Rational. Основные модели и диаграммы UML
10. Экстремальное программирование. Принципы «живой» разработки ПО
11. Анализ предметной области. Схема Захмана.
12. Диаграммы потоков данных. Диаграммы сущностей и связей
13. Выделение и анализ требований. Потребности, функции и требования к ПО
14. Варианты использования и действующие лица. Диаграммы вариантов использования.
15. Качество ПО. Набор стандартов ISO для оценки качества разработки ПО Стандарт качества технологических процессов разработки ПО.
16. Методы контроля качества ПО. Верификация, валидация, тестирование. Ошибки ПО.
17. Архитектура ПО, компоненты и представление архитектуры, методы оценки архитектуры,
18. Виды UML диаграмм. Диаграммы классов, сценариев, компонентов, Диаграммы взаимодействия и развёртывания
19. Образцы проектирования и их классификация. Шаблоны образцов проектирования.
20. Примеры образцов анализа и архитектурных стилей: образец анализ «величина», архитектурные стили «каналы и фильтры», «многоуровневая система, «данные–представление–обработка»
21. Образец проектирования – идиома «шаблонный метод»
22. Образцы организации и образцы процессов, инспекция программ по Фагану
23. Объектно-ориентированная разработка ПО. Инкапсуляция, наследование, полиморфизм.
24. Объектно-ориентированные языки программирования.
25. Агрегация, обобщение, наследование.
26. Объекты, классы, методы, операторы, перегрузка.

27. Область видимости. Статические переменные и функции, статические методы и члены данных классов.
28. Спецификаторы доступа в классах. Спецификаторы доступа при наследовании.
29. Абстрактные и конкретные классы, множественное наследование.
30. Выявление асинхронного параллелизма.
31. Распределение модулей и подсистем по процессам и задачам.
32. Управление глобальными ресурсами и программным обеспечением.