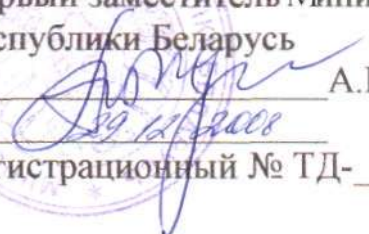


**Министерство образования Республики Беларусь
Учебно-методическое объединение вузов Республики Беларусь
в области информатики и радиоэлектроники**

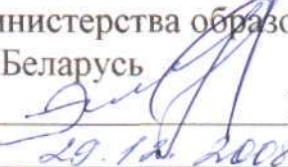
УТВЕРЖДАЮ
Первый заместитель Министра образования
Республики Беларусь

_____ А.И. Жук
_____ 29.12.2008
Регистрационный № ТД- I. 101 /тип.

**ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ
И ПРОЕКТИРОВАНИЕ**

Типовая учебная программа для высших учебных заведений
по специальности
1-53 01 02 Автоматизированные системы обработки информации


СОГЛАСОВАНО
Начальник Управления
радиоэлектронной промышленности
Министерства промышленности
Республики Беларусь

_____ Э.Ф. Лобанович

СОГЛАСОВАНО
Начальник Управления высшего и
среднего специального образования
Министерства образования Респуб-
лики Беларусь

_____ Ю.И. Миксюк
_____ 29.12.2008

Председатель УМО вузов
Республики Беларусь
по образованию в области
информатики и радиоэлектроники

_____ М.Л. Батура
_____ 01.10.2008

Первый проректор Государственного
учреждения образования
«Республиканский институт высшей
школы»

_____ И. В. Казакова
_____ 16.12.2008

Эксперт-нормоконтролер

_____ С.М. Артемьева
_____ 16.12.2008
 

Минск 2008

СОСТАВИТЕЛЬ:

М.П. Ревотнж, профессор кафедры информационных технологий автоматизированных систем Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», доцент, кандидат технических наук

РЕЦЕНЗЕНТЫ:

Кафедра вычислительной техники Учреждения образования «Белорусский государственный аграрный технический университет» (протокол №2 от 25.09.2008);

М.К. Буза, профессор кафедры математического обеспечения ЭВМ Белорусского государственного университета, доктор технических наук, профессор

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой информационных технологий автоматизированных систем Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол №2 от 22.09.2008 г.);

Научно-методическим советом Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол №1 от 24.09.2008);

Научно-методическим советом по направлению I-53 Автоматизация УМО вузов Республики Беларусь по образованию в области информатики и радиоэлектроники (протокол №1 от 15.09.2008)

Ответственный за выпуск: Ц. С.Шикова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Типовая учебная программа «Объектно-ориентированное программирование и проектирование» разработана для студентов высших учебных заведений по специальности 1-53 01 02 Автоматизированные системы обработки информации в соответствии с требованиями Образовательного стандарта ОСРБ 1-53 01 02-2007 и типового учебного плана специальности 1-53 01 02 Автоматизированные системы обработки информации.

Целью изучения курса является углубленное обучение студентов технологическим основам и практическим навыкам проектирования, реализации и сопровождения больших программных систем современных ЭВМ на основе технологии объектно-ориентированного программирования.

Основные задачи изучения дисциплины:

- формирование представления об объектной технологии как примере использования системного подхода и реализации результатов системного анализа;
- приобретение знаний о возможностях, методах, моделях и средствах поддержки современных промышленных информационных технологий;
- приобретение навыков практической работы со средствами обеспечения жизненного цикла создания и эволюционного развития сложных программных систем.

Для изучения курса «Объектно-ориентированное программирование и проектирование» необходимы базовые знания по следующим дисциплинам: «Основы алгоритмизации и программирования», «Математические модели информационных процессов и управления», «Базы и банки данных», «Системное программное обеспечение».

В результате изучения дисциплины обучаемый должен:

знать:

- принципы объектно-ориентированного программирования;
- способы реализации отношений между классами;
- использование свойств полиморфизма, наследования и инкапсуляции;
- возможности и ограничения абстрактных классов, интерфейсов и шаблонов;

уметь:

- создавать структурированные программы на основе объектных технологий в среде современных систем объектно-ориентированного проектирования;
- переходить из одной объектно-ориентированной платформы на другую;
- использовать возможности классов и языка UML для представления проектных решений.

Программа рассчитана на объем 458 учебных часов, из них 204 - аудиторных. Примерное распределение аудиторных часов по видам занятий: лекций - 136 часов, лабораторных работ - 52 часа, практических занятий - 16 часов.

Примерный тематический план дисциплины

Наименование раздела, темы	Всего аудит. часов	Лек- ции, ч	Лабора- торные занятия, ч	Практи- ческие занятия, ч
Введение	2	2		
Раздел 1. Объектно-ориентированное программирование	100	66	34	
Тема 1. Обзор языка программирования C++	12	8	4	
Тема 2. Классы как типы объектов	12	8	4	
Тема 3. Определение операций над объектами классов	6	2	4	
Тема 4. Производные классы	12	8	4	
Тема 5. Объектно-ориентированный ввод — вывод	10	6	4	
Тема 6. Особенности объектно-ориентированного программирования	4	4		
Тема 7. Шаблоны функций и классов	12	8	4	
Тема 8. Обработка исключений	10	6	4	
Тема 9. Динамическая идентификация и приведение типа	8	6	2	
Тема 10. Системы объектно-ориентированного программирования	14	10	4	
Раздел 2. Объектно-ориентированное проектирование	102	68	18	16
Тема 11. Жизненный цикл программных систем	8	4		4
Тема 12. Методы моделирования требований	10	8	2	
Тема 13. Описание динамики системы	10	6	4	
Тема 14. Описание данных системы	10	6		4
Тема 15. Назначение и основные понятия языка UML	10	6	4	
Тема 16. Функциональные требования и диаграммы вариантов использования	10	6	4	
Тема 17. Проектирование внутренней структуры приложений	10	6		4
Тема 18. Проектирование динамики приложений	10	6	4	
Тема 19. Управление требованиями к программному продукту	6	6		

1	2	3	4	5
Тема 20. Методы обеспечения качества разработки программ	16	12		4
Заключение	2	2		
Итого:	204	136	52	16

СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

ВВЕДЕНИЕ

Цель и задачи дисциплины, ее роль и место в общей системе подготовки специалиста. Связь курса с другими дисциплинами специальности.

Виды технологий разработки сложных программ. Структурное и модульное программирование. Синтезирующие и конкретизирующие технологии. Процедурное и объектно-ориентированное программирование. Базовые понятия технологии объектно-ориентированного программирования: абстракция данных, пакетирование, наследование, полиморфизм. Обзор средств объектно-ориентированного программирования.

Раздел 1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Тема 1. ОБЗОР ЯЗЫКА ПРОГРАММИРОВАНИЯ C++

Характеристика языка программирования C++. Соотношение языков C и C++. Комментарии в языке C++. Идентификаторы объектов. Область действия (видимости) объекта. Типы объектов. Атрибуты типа `const` и `volatile`. Ссылки (обращение по адресу). Набор и приоритет операций в языке C++. Операция преобразования(приведения) типа. Управление размещением объектов в памяти. Объявление и переопределение функций. Установка умалчиваемых значений параметров функций. Функции с переменным числом параметров. Функции с подстановкой тела.

Тема 2. КЛАССЫ КАК ТИПЫ ОБЪЕКТОВ

Понятие класса объектов. Виды классов. Синтаксис описания и определения класса. Схема взаимосвязи элементов класса и внешних объектов. Атрибуты доступа к элементам класса. Функции-элементы и элементы данных класса. Статические элементы класса. Дружественные функции класса. Дружественные классы. Конструкторы и деструкторы объектов. Особенности связи с конструкторами и деструкторами. Правила надежного определения класса. Примеры определения классов. Особенности образования структурированных объектов произвольных классов. Конструкторы вложенных классов. Массивы объектов классов. Статические объекты произвольных классов.

Тема 3. ОПРЕДЕЛЕНИЕ ОПЕРАЦИЙ НАД ОБЪЕКТАМИ КЛАССОВ

Схема определения операций над объектами. Особенности определения операций. Способы согласования типов объектов. Особенности использования ссылочных типов. Примеры определения операций над классами.

Особенности операций присваивания и инициализации. Переопределение операции индексации. Переопределение операции вызова функции. Переопределение операций управления памятью.

Тема 4. ПРОИЗВОДНЫЕ КЛАССЫ

Понятие производного класса. Базовый класс и атрибуты его доступа. Иерархия производных классов. Множественное наследование. Виртуальные базовые классы. Конструкторы производных классов. Указатели объектов производного и базовых классов. Виртуальные функции. Виртуальные деструкторы. Абстрактные классы.

Тема 5. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ВВОД - ВЫВОД

Классы и потоки ввода-вывода. Стандартные потоки ввода-вывода. Схема иерархии классов ввода-вывода. Ввод-вывод объектов базовых типов. Примеры программ ввода-вывода с использованием стандартных потоков.

Ввод-вывод объектов определенных пользователем классов. Контроль исключительных ситуаций ввода-вывода. Форматный ввод-вывод. Бесформатный ввод-вывод. Функции-манипуляторы.

Управление позиционированием потока. Связанные потоки. Создание и организация взаимодействия потоков. Буферы потоков. Файловый ввод-вывод. Строчный ввод-вывод.

Тема 6. ОСОБЕННОСТИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Обзор операционных особенностей объектов класса. Особенности динамического управления памятью. Использование статических элементов класса. Статическое и динамическое связывание. Динамические особенности операторов присваивания.

Правила программирования на языке C++.

Тема 7. ШАБЛОНЫ ФУНКЦИЙ И КЛАССОВ

Назначение и виды шаблонов в языке C++.

Шаблоны функций. Определение и использование шаблона функции. Переопределение шаблонов функций и их специализация.

Шаблоны классов. Определение и использование шаблона класса. Переопределение шаблонов классов и их специализация.

Параметры шаблонов.

Особенности использования шаблонов в многофайловых проектах.

Стандартные библиотеки шаблонов STL и ATL.

Характеристика шаблонов определений функций и классов.

Тема 8. ОБРАБОТКА ИСКЛЮЧЕНИЙ

Понятие исключения. Виды и спецификация исключений. Порождение и перехват исключений. Контролируемые блоки и ловушки исключений.

Структурное управление исключениями. Кадрованное и завершающее управление исключениями.

Примеры предопределенных классов исключений. Иерархическое управление исключениями.

Тема 9. ДИНАМИЧЕСКАЯ ИДЕНТИФИКАЦИЯ И ПРИВЕДЕНИЕ ТИПА

Динамическая идентификация типа. Недостатки традиционных операций приведения типа. Обзор новых возможностей приведения типа. Динамическое приведение типа.

Полиморфизм и приведение типа.

Нисходящее приведение типа. Перекрестное приведение типа. Статическое приведение типа. Преобразования типа с сохранением значений.

Тема 10. СИСТЕМЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Обзор систем объектно-ориентированного программирования. Критический анализ средств поддержки ключевых понятий объектно-ориентированного программирования. Область применимости технологии объектно-ориентированного программирования.

Системы быстрой разработки приложений. Системы визуального программирования. Системы программирования фирм Borland и Microsoft.

Библиотечные классы абстрактных структур данных. Иерархия классов абстрактных структур данных. Обзор предопределенных возможностей классов абстрактных структур данных.

Классы итераторов ассоциаций объектов. Компонентные классы. Обзор библиотек MFC и VCL.

Языки программирования Java и C# и их характеристика.

Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ

Тема 11. ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНЫХ СИСТЕМ

Программа и программная система как технологический объект. Концепция программной системы как изделия. Понятие жизненного цикла программных систем.

Модели жизненного цикла программных систем: поэтапная, каскадная, спиральная и другие.

Возможности формализации и автоматизации отдельных стадий и этапов жизненного цикла.

Тема 12. МЕТОДЫ МОДЕЛИРОВАНИЯ ТРЕБОВАНИЙ

Моделирование функций программного обеспечения. История возникновения и развития подходов, связанных с графическим моделированием деятельности. CASE-средства. Подход к моделированию функциональности систем Structured Analysis and Design Technique (SADT). Семейство стандартов средств моделирования Integration DEFinition(IDEF). Стандарт IDEF0. Графические символы стандарта. Виды связей. Правила декомпозиции.

Тема 13. ОПИСАНИЕ ДИНАМИКИ СИСТЕМЫ

Принципы моделирования динамики систем. Методы моделирования динамики систем. Сети Петри, конечные автоматы и диаграммы переходов. Имитационное моделирование систем. Представление поведения системы в терминах событий, процессов и работ. Поток работ.

Модели семейства PME (property-method-event).

Назначение и основные принципы нотации IDEF. Нотация IDEF3. Графические символы IDEF3.

Нотация IDEF4 и объектно-ориентированное проектирование.

Тема 14. ОПИСАНИЕ ДАННЫХ СИСТЕМЫ

Принципы моделирования данных. Иерархические, реляционные и сетевые модели. Нотация Чена - Баркера. Словарь сущностей и атрибутов. Словарь стрелок модели.

Нотация IDEF1x - назначение, основные принципы. Виды сущностей в IDEF1X. Виды связей в IDEF1X.

Процедура нормализации в реляционных моделях. Соответствие между логической и физической моделями данных.

Тема 15. НАЗНАЧЕНИЕ И ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА UML

Язык UML. CASE-средство Rational Rose. Предыстория. Математические основы (теория множеств, теория графов, семантические сети). Диаграммы структурного анализа.

Основные этапы развития UML. Назначение языка UML.

Общая структура языка UML.

Базовые понятия метамодели языка UML (основные элементы, элементы ядра, вспомогательные элементы, механизмы расширения, типы данных, элементы поведения, общее поведение, кооперации, варианты использования, автоматы, общие механизмы, управление моделями).

Специфика представления метамодели языка UML. Особенности изображения диаграмм языка UML.

Тема 16. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ И ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ

Графические элементы модели прецедентов: варианты использования, актеры, интерфейсы, примечания. Отношения на диаграмме вариантов

использования (ассоциации, расширения, обобщения, включения). Рекомендации по разработке диаграмм вариантов использования в UML.

Тема 17. ПРОЕКТИРОВАНИЕ ВНУТРЕННЕЙ СТРУКТУРЫ ПРИЛОЖЕНИЙ

Класс в UML (имя класса, атрибуты класса, операции класса). Отношения между классами (зависимости, ассоциации, агрегации, композиции, обобщения). Интерфейсы. Объекты. Шаблоны или параметризованные классы. Рекомендации по построению диаграмм классов.

Тема 18. ПРОЕКТИРОВАНИЕ ДИНАМИКИ ПРИЛОЖЕНИЙ

Диаграмма состояний (statechart diagram): автоматы, определение состояния и перехода, составное состояние и подсостояния, сложные переходы.

Диаграмма деятельности (activity diagram): состояние действия, переходы, дорожки, объекты. Рекомендации по построению диаграмм деятельности.

Диаграмма последовательности (sequence diagram): объекты, сообщения, рекомендации по построению диаграмм последовательности.

Диаграмма кооперации (collaboration diagram): кооперация, диаграмма кооперации уровня спецификации, объекты, связи, сообщения. Рекомендации по построению диаграмм кооперации.

Тема 19. УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ К ПРОГРАММНОМУ ПРОДУКТУ

Системы управления требованиями. Определение принципов создания хранилищ и политики доступа к ним. Фиксация инструментария, рабочей среды, средств разработки и внешних интерфейсов. Идентификация требований. Определение трассировки требований. Определение группы, видов и атрибутов требований. Определение метрик и отчетов.

Определение принципов управления требованиями, формирование группы управления изменениями. Определение контрольных точек проекта. Внедрение системы управления требованиями. Подсистема управления требованиями на основе IBM Rational RequisitePro.

Тема 20. МЕТОДЫ ОБЕСПЕЧЕНИЯ КАЧЕСТВА РАЗРАБОТКИ ПРОГРАММ

Тестирование приложений. Функциональное и нагрузочное тестирование. CASE-средства Rational Functional Tester, Rational Performance Tester. Виды испытаний и тестирования на различных этапах разработки. Критерии оценки качества систем различного назначения. Документирование испытаний.

Возможности объектно-ориентированной технологии тестирования. Место тестирования в жизненном цикле разработки программ. Фазы тестирования. Общая схема тестирования. Роли и ответственность. Методы тестирования («белый ящик», «серый ящик», «черный ящик»). Статические проверки. Планирование тестирования. Функциональное тестирование с использованием инструментальных средств IBM Rational.

Нагрузочное тестирование. Основы использования инструментальных средств IBM Rational Software в нагрузочных испытаниях программного обеспечения. Обзор IBM Rational Suite TestStudio и IBM Rational Administrator. Разработка расписаний. Выполнение расписаний. Анализ результатов тестирования. Язык описания виртуальных пользователей. Корреляция данных. Тестирование больших систем. Инсталляция и конфигурация тестового окружения.

Обеспечение информационного обмена между группами в рамках проектных команд, сокращение риска возникновения ошибок. Технологический и управленческий комплекс Rational Suite.

ЗАКЛЮЧЕНИЕ

Оценка качества программных систем. Факторы качества, критерии качества, метрики и оценочные элементы. Методы оценки: расчетный, экспертный. Взаимосвязь факторов, критериев и метрик с фазами жизненного цикла программных систем. Выбор оценочных элементов для программных систем различного назначения.

Методы управления качеством, используемые в современных технологиях программирования. Аттестация программных систем. Инструментальные системы оценки качества программных систем.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ РАБОТ

1. Расширение возможностей языка C в языке C++
2. Программирование в терминах классов
3. Манипулирование объектами классов
4. Производные и абстрактные классы
5. Техника работы со библиотеками классов
6. Объектно-ориентированный ввод-вывод
7. Использование шаблонов функций и классов
8. Обработка исключений в C++
9. Проектирование визуальных компонент
10. Инструментальные средства поддержки UML
11. Проектирование на основе прецедентов
12. Обратное проектирование и UML
13. Тестирование программных комплексов

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ НЕОБХОДИМОГО ОБОРУДОВАНИЯ И КОМПЬЮТЕРНЫХ ПРОГРАММ

1. Класс современных персональных ЭВМ.
2. Система программирования Microsoft Visual C++ 6.x и выше
3. Система программирования Inprise C++ Builder 5.x и выше
4. Система электронной справочной документации MSDN Library
5. Система программирования Java SDK 6.x и выше

6. Программные комплексы IBM ANFusion, Rational Suite.

ЛИТЕРАТУРА

ОСНОВНАЯ

1. Буч Г. Объектно-ориентированное проектирование с примерами применения: Пер с англ. - СПб.: Невский Диалект, 2001. - 560 с.
2. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. - СПб.: Питер, 2001.-386 с.
3. Ларман К. Применение UML 2.0 и шаблонов проектирования: Пер с англ. - М.: Вильямс, 2007. - 736 с.
4. Страуструп Б. Язык программирования C++. Специальное издание: Пер с англ.- М.: БИНОМ, 2007. - 1104 с.
5. Шилдт Г. Полный справочник по C++: Пер с англ. - М.: Вильяме, 2003.- 800 с.

ДОПОЛНИТЕЛЬНАЯ

6. Голуб А.И. С и C++. Правила программирования: Пер с англ. - М.: БИНОМ, 1997.-272 с.
7. Лафоре Р. Объектно-ориентированное программирование в C++: Пер с англ. - СПб.: Питер, 2007. - 928 с.
8. Калверт Ч. Borland C++ Builder. Энциклопедия пользователя: Пер с англ. - К.: ДИАСОФТ, 1998. - 830 с.
9. Мюллер Дж. Visual C++ 5/Пер с англ. - СПб.: ВHV - Санкт-Петербург, 1998.-720 с.
10. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#: Пер с англ. - СПб.: Питер, 2008. - 856 с.
11. Корнелл Г., Хорстманн К. Java 2. Том I. Основы. Библиотека профессионала: Пер с англ. - М.: Вильяме, 2008. - 816 с.
12. Седжвик Р. Фундаментальные алгоритмы на C++. Части 1-4. Анализ. Структуры данных. Сортировка. Поиск: Пер с англ. - К.: ДиаСофт, 2002. - 687 с.
13. К. Ш. Тан, В.-Х. Стиб, И. Харди, Символьный C++. Введение в компьютерную алгебру с использованием объектно-ориентированного программирования. - М.: Мир, 2001. - 624 с.
14. Иванова Г. С. Ничушкина Т. Н. Пугачев Е. К., Объектно-ориентированное программирование: Учебник для вузов. - М.: Издательство Московского государственного технического университета, 2001. - 320 с.
15. Йордон Э., Аргила К. Структурные модели в объектно-ориентированном анализе и проектировании. - М.: Лори, 1999. - 268 с.