

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по естественнонаучному образованию

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

В.А.Богуш

2016 г.

Регистрационный № ТД- 6.580 /тип.



ПРОЕКТИРОВАНИЕ ПРОГРАММНЫХ СИСТЕМ

Типовая учебная программа по учебной дисциплине

для направления специальности

1- 31 03 07 – 01 Прикладная информатика

(программное обеспечение компьютерных систем)

СОГЛАСОВАНО

Председатель Учебно-методического объединения по естественнонаучному образованию

А.П. Толстик

2015 г.



СОГЛАСОВАНО

Начальник Управления высшего образования Министерства образования Республики Беларусь

С.И. Романюк

2016 г.

« 3 » мая

СОГЛАСОВАНО

Проректор по научно-методической работе Государственного учреждения образования «Республиканский институт высшей школы»

И.В. Титович

2016 г.



Эксперт-нормоконтролер

В.П. Швабко

2016 г.

« 06 »

04

Минск 2016

СОСТАВИТЕЛИ:

А.Н. Курбацкий, заведующий кафедрой технологий программирования Белорусского государственного университета, доктор технических наук, профессор;

М.И. Давидовская, старший преподаватель кафедры технологий программирования Белорусского государственного университета.

РЕЦЕНЗЕНТЫ:

Кафедра программного обеспечения информационных технологий Учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

И.А. Король, заместитель директора республиканского унитарного предприятия «Национальный центр электронных услуг», кандидат физико-математических наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой технологий программирования Белорусского государственного университета
(протокол № 10 от 19.03.2015 г.);

Научно-методическим советом Белорусского государственного университета
(протокол № 6 от 29.06.2015 г.);

Научно-методическим советом по прикладной математике и информатике учебно-методического объединения по естественнонаучному образованию
(протокол № 10 от 2.06.2015 г.).

Ответственный за редакцию: М.И. Давидовская

Ответственный за выпуск: М.И. Давидовская

Пояснительная записка

Типовая учебная программа по учебной дисциплине «Проектирование программных систем» разработана в соответствии с типовым учебным планом и образовательным стандартом высшего образования первой степени по направлению специальности 1 – 31 03 07 – 01 «Прикладная информатика (программное обеспечение компьютерных систем)».

Учебная дисциплина «Проектирование программных систем» знакомит студентов с технологиями моделирования и разработки программных систем (ПС).

Основой для изучения проектирования программных систем являются учебные дисциплины «Программирование» и «Операционные системы».

Учебная дисциплина «Проектирование программных систем» связана с учебной дисциплиной «Модели данных и системы управления базами данных» и другими общепрофессиональными учебными дисциплинами, предусмотренными типовым учебным планом по специальности. Методы проектирования приложений, излагаемые в учебной дисциплине «Проектирование программных систем», являются основой для изучения дисциплин «Проектирование человеко-машинных интерфейсов», «Программирование мобильных систем», «Web-программирование», «Менеджмент программного обеспечения», а также при изучении ряда дисциплин специализации.

Цель преподавания учебной дисциплины «Проектирование программных систем»: дать систему знаний по теоретическим и прикладным основам проектирования и разработки программных систем.

Основные задачи, решаемые при изучении учебной дисциплины «Проектирование программных систем»:

- изучение основ моделирования, проектирования и разработки программных систем;
- изучение управления программными проектами (от инициации до реализации и внедрения);
- освоение работы с различными методологиями, технологиями и платформами разработки программного обеспечения.

В результате изучения учебной дисциплины студент должен

знать:

- архитектуру программных систем и подходы к ее проектированию;
- концепции теории управления проектами (критерии успешности проекта, организацию проектной команды, жизненный цикл проекта);
- организационно-методическое обеспечение проекта (приоритеты, цели и результаты, допущения и ограничения, ресурсы, сроки, риски, критерии приемки);
- вопросы планирования проекта, а так же базовое расписание проекта;
- риски программного проекта (качественный и количественный анализ рисков, главные риски и способы реагирования, мониторинг и контроль рисков, планирование реагирования на риски);

- вопросы оценки трудоемкости и сроков разработки программного обеспечения;

- вопросы формирования команды реализации проекта, а также этапы реализации, внедрения и сопровождения проекта;

уметь:

- исследовать структуру и характеристики автоматизированной (программной) системы;

- строить концептуальную модель предметной области проектируемой программной системы;

- разрабатывать техническое задание на программную систему;

- строить диаграмму вариантов использования, диаграмму деятельности, диаграмму последовательности и диаграмму классов проектируемого программного обеспечения;

- генерировать код, отлаживать и тестировать программное обеспечение;

владеть:

- технологиями моделирования программных систем;

- технологиями проектирования и разработки программных систем;

- CASE-технологиями проектирования автоматизированных систем.

Требования к академическим компетенциям специалиста

Специалист должен:

АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

АК-2. Владеть системным и сравнительным анализом.

АК-3. Владеть исследовательскими навыками.

АК-4. Уметь работать самостоятельно.

АК-5. Быть способным вырабатывать новые идеи (креативность).

АК-6. Владеть междисциплинарным подходом при решении проблем.

Требования к профессиональным компетенциям специалиста

Специалист должен быть способен:

Проектно-конструкторская деятельность

ПК-1. Проектировать, разрабатывать и тестировать программное обеспечение различных видов.

ПК-2. Разрабатывать техническую документацию на программное обеспечение.

ПК-3. Проектировать, разрабатывать и тестировать техническое обеспечение компьютерных и телекоммуникационных систем различных видов.

Экспертно-консультационная деятельность

ПК-18. Оказывать консультации по вопросам работы программного обеспечения, в том числе разработанного сторонними организациями.

ПК-21. Анализировать результаты работы установленного программного обеспечения и вырабатывать предложения по улучшению качества его работы.

ПК-23. Проводить обучение специалистов, занимающихся эксплуатацией программного обеспечения.

Типовая учебная программа рассчитана на 104 часа, из них 68 аудиторных часов, примерное распределение которых по видам занятий включает 34 лекционных часа и 34 часа лабораторных занятий.

Примерный тематический план

№	Наименование раздела, темы	Количество аудиторных часов	
		Лекции	Лабораторные занятия
1	1. Основы программной инженерии	2	
2	Раздел I. Архитектура и проектирование программных систем	20	30
3	2. Архитектура программных систем и моделирование	2	
4	3. Унифицированный язык моделирования UML	4	8
5	4. Требования к ПО: их определение, классификация и управление	4	6
6	5. Анализ и проектирование программного обеспечения	4	12
7	6. Проектирование баз данных	2	4
8	7. Объектный язык ограничений OCL	2	
9	8. Шаблоны (образцы) проектирования	2	
10	Раздел II. Управление программными проектами и обеспечение качества	6	4
11	9. Управление ресурсами, проектами, рисками	1	
12	10. Управление дефектами	2	2
13	11. Совместная работа над проектами. Системы контроля версий	1	2
14	12. Сборка и выпуск программных проектов. Непрерывная интеграция ПО	1	
15	13. Качество программного обеспечения	1	
16	Раздел III. Методологии проектирования программных систем и управления ИТ-услугами	6	
17	14. Методологии управления ИТ-услугами	2	
18	15. Методологии разработки программного обеспечения	4	
19	Всего часов	34	34
		68	

Содержание учебного материала

1. Основы программной инженерии

Системный подход как основа проектирования программного обеспечения. Классификации программного обеспечения и программных проектов. Особенности современных программных проектов. Особенности развития компьютерной индустрии. Этапы становления и развития программной инженерии. Жизненный цикл программного обеспечения (ПО). Процессы жизненного цикла согласно ISO 12207:1995 и ISO 12207:2008. Модель жизненного цикла (ЖЦ), стадия, контрольная точка (веха). Модели ЖЦ: каскадная, эволюционная, основанная на формальных преобразованиях, пошаговая итерационная, спиральная. Сравнение разных моделей между собой, их достоинства и недостатки. Методологии разработки ПО: краткий обзор. Введение в методологию гибкой разработки.

Раздел I. Архитектура и проектирование программных систем

2. Архитектура программных систем и моделирование

Понятие архитектуры ПО. Архитектурные представления. Модель «4+1». Гибкое моделирование, характеристики «гибкой» модели, ценности гибкого моделирования, принципы гибкого моделирования.

Принципы построения объектной модели (по Гр. Бучу). Основные элементы объектной модели: объект, индивидуальность, поведение объекта, состояние объекта, класс, атрибут, операция, кооперация, компонент, интерфейс, пакет и подсистема. Виды связей между элементами моделей: соединение, ассоциация, агрегация, композиция, зависимость, реализация, обобщение. Понятие полиморфизма. Направление ассоциации, мощности атрибутов и полюсов. Характеристики полюсов ассоциаций с мощностью «*» (set, ordered set, bag, sequence). Квалификаторы. N-арные ассоциации ($N > 2$), классы ассоциаций. Определения и примеры. Общие свойства и различия пакетов и подсистем.

3. Унифицированный язык моделирования UML

Унифицированный язык моделирования (Unified Modeling Language – UML), спецификации и история развития. Классификация диаграмм для версий UML 1.x и UML 2.x. SysML как расширение языка UML.

Диаграммы вариантов использования, их элементы и связи между элементами. Правила обозначения вариантов использования, действующих лиц. Связи ассоциации и коммуникации. Связи включения и расширения. Связь обобщения. Правила составления спецификаций и сценариев. Дополнительные обозначения языка UML для бизнес-моделирования.

Диаграммы деятельности, их элементы и связи. Разделы, сигналы, и области расширения. Область применения диаграмм деятельности.

Диаграммы состояний, их элементы и связи. Область их применения.

Диаграммы взаимодействия, их элементы и связи между элементами. Виды диаграмм взаимодействия и область их применения. Типы сообщений на диаграммах последовательности, комбинированные фрагменты взаимодействия, операторы взаимодействия.

Коммуникационные диаграммы, их элементы и связи. Область их применения.

Диаграммы обзора взаимодействия, их элементы и связи. Область их применения.

Временные диаграммы, их элементы и связи. Область их применения.

Структурные диаграммы. Диаграммы классов, их элементы и связи между элементами. Статические операции и атрибуты. Интерфейсы и абстрактные классы. Шаблон класса (параметризованный класс). Активный класс. Область применения диаграмм классов.

Диаграммы объектов, их элементы и связи. Область их применения.

Диаграммы пакетов, их элементы и связи. Область их применения.

Диаграммы компонентов и диаграммы размещения, их элементы и связи между элементами. Область их применения.

Общая иерархия моделей (M0, M1, M2, M3). Назначение метамодели UML, её структура. Механизмы расширения UML (стереотипы, метасвойства, ограничения, комментарии).

Диаграммы профиля, их элементы и связи. Область их применения.

4. Требования к ПО: их определение, классификация и управление

Понятие требования к программному обеспечению. Виды требований. Классификация требований по SWEBOOK. Классификация требований по Вигерсу. Классификация требований по RUP. Классификация требований по IEEE 830. Другие классификации требований.

Выявление и анализ требований. Документирование и организация требований. Процесс определения требований, его цели, содержание, исполнители и рабочие продукты. Варианты использования. Описание варианта использования. Виды сценариев вариантов использования. Область действия варианта использования. Уровень цели варианта использования. Модель вариантов использования, ее элементы, связи, диаграммы. Параметризованные варианты использования и варианты использования CRUD. Изменение и версионирование требований. Трассировка требований.

5. Анализ и проектирование программного обеспечения

Анализ программного продукта. Процесс анализа и проектирования в методологии RUP, задействованные в нём исполнители, рабочие продукты, виды выполняемых работ. Архитектурный анализ, его цели, содержание, исполнители и рабочие продукты. Соглашения моделирования.

Механизмы анализа. Идентификация ключевых абстракций. Формирования архитектурных уровней. Понятие шаблона (образца) и способ его описания. Шаблоны анализа: «уровни», «модель-представление-управление», «каналы и фильтры». Анализ вариантов использования, его цели и содержание, исполнители и рабочие продукты.

Идентификация классов анализа. Определение атрибутов и ассоциаций классов анализа.

Шаблоны (образцы) распределения обязанностей между классами (GRASP). Шаблоны «Сценарий транзакции» и «Модель предметной области». Унификация классов анализа.

Проектирование программного продукта. Проектирование архитектуры системы, его цели и содержание, исполнители и рабочие продукты. Выявление проектных классов, пакетов, подсистем и интерфейсов.

Проектные механизмы и механизмы реализации. Проектирование структуры потоков управления. Цели, содержание, исполнители и рабочие продукты. Примеры диаграмм классов, моделирующих потоки управления. Активные классы. Проектирование конфигурации системы. Цели, содержание, исполнители и рабочие продукты. Примеры диаграмм размещения. Проектирование классов, его цели и содержание (детализация проектных классов, уточнение операций классов, моделирование состояний, уточнение атрибутов классов, уточнение связей между классами), исполнители и рабочие продукты. Выводимые атрибуты и ассоциации. Примеры

6. Проектирование баз данных

Модели баз данных. Объектно-реляционное отображение. Отображение классов. Отображение атрибутов, в том числе атрибутов с мощностью более 1, производных и статических атрибутов. Стратегии отображения бинарных ассоциаций: слияние таблиц, добавление внешнего ключа, отдельная таблица для связи. Отображение квалифицированных ассоциаций, n-арных ($n > 2$) ассоциаций, классов ассоциаций. Стратегии отображения обобщения: единая таблица для всей иерархии, таблицы для каждого класса, таблицы только для конкретных классов, таблицы только для различных конкретных классов. Профиль для моделирования схем баз данных диаграммами классов. Универсальный мэппинг С. Амблера.

7. Объектный язык ограничений OCL

Объектный язык ограничений язык (Object Constraint Language – OCL). Основные понятия, элементы выражений, навигация по ассоциациям, коллекции, операции над коллекциями (collect, select, forAll, exists, isEmpty, notEmpty, size, sum, min, max, count, includes, excludes, intersection, union, closure, flatten, sortBy). Операции упорядоченных коллекций (append, prepend, insertAt, indexOf, at, first, last, reverse). Конструкции let, iterate. N-ки (tuple), операция product. Операции allInstances, oclIsTypeOf, oclIsKindOf. Приведение типов oclAsType, asSet, asBag. Наследование ограничений.

8. Шаблоны (образцы) проектирования

Понятие шаблона (образца) и способ его описания. Шаблоны проектирования (абстрактная фабрика, фабричный метод, адаптер, компоновщик, мост, фасад, заместитель, цепочка обязанностей, итератор, стратегия, декоратор, наблюдатель). Примеры использования шаблонов (образцов) проектирования.

Раздел II. Управление программными проектами и обеспечение качества

9. Управление ресурсами, проектами, рисками

Понятие ресурса. Виды ресурсов. Сотрудники и роли. Процессы и роли. Роли в процессе разработки программных проектов. Совмещение ролей. Связи между ролями. Управление прочими ресурсами.

Программные проекты. Задачи. Изменения в проекте. Исправления программных дефектов. Этап программного проекта. Вехи проекта. Выполнение

проекта. Визуализация плана с помощью сетевого графика и диаграммы Ганта. Наблюдение за проектом.

Риски в программных проектах. Управление рисками. Идентификация рисков. Анализ и ранжирование рисков. Планирование управления риском. Оценка рисков и уровни влияния на стоимость, сроки, технические показатели программных проектов. Матрица риска. Методика управления рисками. Стандарты управления рисками.

10. Управление дефектами

Дефекты. Характеристики дефектов: идентификатор, содержание, срочность, категория, серьезность, автор и т.д. Жизненный цикл дефекта. Системы управления дефектами. Нотификации в системах управления дефектами. Основные свойства систем управления дефектами.

11. Совместная работа над проектами. Системы контроля версий

Групповая работа над проектом. Общий доступ к файлам и хранение истории модификации файлов. Поддержание и развитие нескольких историй файла. Системы контроля версий в программных проектах. Типы систем контроля версий. Ревизия файла и ее атрибуты. Хранимые копии проекта. Общие принципы хранения файлов в системах версионирования. Основные способы пометки версий в СКВ. Операции в системах контроля версий. Разрешение конфликтов. Примеры систем контроля версий (CVS, Subversion, Git и др.)

12. Сборка и выпуск программных проектов. Непрерывная интеграция ПО

Понятия сборки и выпуска программных проектов. Задачи сборки и выпуска проектов. Причины и проблемы сборки проектов. Окружение для сборки. Общие требования к сборке проектов. Способы сборки программных проектов. Требования к процедурам сборки. Утилиты и средства сборки проектов.

Понятие непрерывной интеграции. Действия в системах непрерывной интеграции. Промышленные средства непрерывной интеграции.

13. Качество программного обеспечения

Тенденции индустрии разработки ПО. Известные примеры программных ошибок. Что такое качественное ПО? Характеристики качества ПО. Стандарты качества ПО. Оценка качества ПО. Метрики программного обеспечения и их использование. Аудит программного кода.

Динамические и статические методы обеспечения качества ПО. Формальная верификация. Метод проверки моделей. Обеспечение качества ПО путем обнаружения ошибок. Динамические и статические методы обнаружения программных дефектов.

Раздел III. Методологии проектирования программных систем и управления ИТ-услугами

14. Методологии управления ИТ-услугами

Методологии управления ИТ-услугами на примере ITSM, ITIL, CobiT, СММІ. Их история, структура, основные концепты и Области усовершенствования. Применений методологий управления ИТ-услугами для разработки программного обеспечения. Уровни зрелости процессов по СММІ.

15. Методологии разработки программного обеспечения

Критерии успешности программного проекта. Методологии проектирования ПО и их характеристики. Стратегии конструирования ПО. Адаптивность процесса разработки ПО к окружению. Как выбрать методологию проектирования? Чем отличаются различные методологии проектирования? Классическая модель проектирования ПО, ее достоинства и недостатки. Прототипирование ПО, достоинство и недостатки. Инкрементная модель, ее достоинства и недостатки. Спиральная модель, ее достоинства и недостатки.

Основные определения. Rational Unified Process (RUP). Основные принципы RUP. Общее представление RUP. Стадии жизненного цикла RUP и их содержание. Контрольные точки в жизненном цикле RUP. Процессы жизненного цикла RUP, их задачи, содержание, основные рабочие продукты и задействованные роли.

Быстрая разработка приложений (Rapid Application Development - RAD). Основные этапы RAD. Бизнес-моделирование и моделирование данных. Генерация приложений. Достоинства и недостатки RAD.

Гибкие (Agile) методологии разработки программного обеспечения. Основные особенности методологии гибкой разработки ПО. Манифест гибкой разработки ПО. Принципы гибкой разработки. Современные Agile- методологии: классификация. Характеристики экстремального программирования (eXtreme Programming — XP). Методы XP. Тестирование, рефакторинг, парное программирование, коллективное владение, стандарты кодирования в XP. Методология SCRUM и ее характеристики.

Методология разработки программного обеспечения Microsoft Solutions Framework (MSF). История и текущий статус. Основные принципы MSF. Модель команды. Другие особенности MSF.

Технология Microsoft Visual Studio Team System (VSTS). Обзор VSTS. Управление элементами работ. Конфигурационное управление. Поддержка различных моделей процесса и тестирование.

Методология разработки программного обеспечения DevOps. История и текущий статус. Основные принципы DevOps. Модели внедрения DevOps.

Информационно-методическая часть

Литература

Основная

1. Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки – СПб.: Питер, 2005. – 416 с.
2. Арлоу Дж., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование: 2-ое изд. – СПб. Символ-Плюс, 2013. – 624 с.
3. Буч Г., Янг Б., Коналлен Дж., Хьюстон К., Энгл М., Максимчук Р. Объектно-ориентированный анализ и проектирование с примерами приложений. 3-е изд. – М.: Вильямс, 2012. – 720 с.

4. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учеб. пособие. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2006. – 192 с.: ил.
5. Вигерс К. Разработка требований к программному обеспечению. Пер. с англ. – М.: Русская редакция, 2004. – 576 с.: ил.
6. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2010. – 368 с.: ил.
7. Коберн А. Быстрая разработка программного обеспечения. – М.: ЛОРИ, 2013. – 314 с.: ил.
8. Коберн А. Современные методы описания функциональных требований к системам.: Пер. с англ. – М. Лори, 2010. – 264 с.: ил.
9. Кролл П., Крачтен Ф. Rational Unified Process – это легко. Руководство по RUP для практиков. – М.: КУДИЦ-ОБРАЗ. 2004. – 432 с.: ил.
10. Кузнецов С. Базы данных. Вводный курс [Электронный ресурс]/ Центр Информационных Технологий. – Москва, 2008. – Режим доступа: http://citforum.ru/database/advanced_intro/32.shtml. – Дата доступа: 18.01.2014 г.
11. Кулямин В. В. Технологии программирования. Компонентный подход. – М.: Бином. Лаборатория знаний, 2007. – 463 с.: ил.
12. Ларман Крэг. Применение UML 2.0 и шаблонов проектирования. 3-е изд. Пер. с англ. – М.: Вильямс, 2007. – 624 с.: ил.
13. Леоненков А. Самоучитель UML (2-е издание). – СПб.: БХВ-Петербург, 2004. – 432 с.: ил.
14. Мацяшек Л. А. Анализ и проектирование информационных систем с помощью UML 2.0 – СПб.: Вильямс, 2008. – 816 с.: ил.
15. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. 2-е изд. – СПб.: Питер, 2006. – 544 с.: ил.
16. Sommerwil I. Инженерия программного обеспечения. 6-е издание.: Пер. с англ.: – М.: Вильямс, 2002. – 626 с.: ил.
17. Фаулер М. UML. Основы. 3-е издание. Краткое руководство по стандартному языку объектного моделирования.: Пер. с англ. – СПб: Символ-Плюс, 2005. – 192 с.: ил.
18. Фаулер М. Архитектура корпоративных программных приложений. – М.: Вильямс, 2006. – 544 с.: ил.
19. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения.: Пер. с англ. – СПб.: Питер, 2002. – 496 с.
20. Warner J., Kleppe A. Object Constraint Language, The: Getting Your Models Ready for MDA, Second Edition. 192 p.

Дополнительная

21. Брукс Ф. Мифический человеко-месяц или как создаются программные системы. – СПб.: Символ-Плюс, 1999. – 288 с.: ил.
22. Буч Г. Язык UML. Руководство пользователя. – М.: ДМК, 2007. – 496 с.
23. Грэхем И. Объектно-ориентированные методы. Принципы и практика. 3-е изд. Пер. с англ. – М.: Вильямс, 2004. – 400 с.: ил.

24. Дубинина О. Обзор паттернов проектирования [Электронный ресурс]/ Центр Информационных Технологий. – Москва, 2005. – Режим доступа: <http://citforum.ru/SE/project/pattern/>. – Дата доступа: 28.01.2014 г.
25. Кериевски Дж. Рефакторинг с использованием шаблонов. – М.: Вильямс, 2006 г. – 400 с.: ил.
26. Коналлен Дж. Разработка Web-приложений с использованием UML.: Пер. с англ. – М.: Вильямс, 2001. – 356 с.: ил.
27. Полис Г., Огастин Л. и др. Разработка программных проектов на основе Rational Unified Process.: Пер. с англ. – М.: Бином-Пресс, 2005. – 544 с.: ил.
28. Фримен Э., Фримен Э. и др. Паттерны проектирования – СПб: Питер, 2011. – 296 с.: ил.
29. Ambler S. Agile Database Techniques – Effective Strategies for the Agile Software Developer [Электронный ресурс]/ Ambysoft Inc. – Toronto, 2003. – Режим доступа: <http://www.agiledata.org/essays/umlDataModelingProfile.html>. – Дата доступа: 12.02.2014 г.
30. OCL portal [Электронный ресурс]/ Technische Universität. – Dresden, 2005. – Режим доступа: <http://www-st.inf.tu-dresden.de/ocl/>. – Дата доступа: 14.02.2014 г.

Методические рекомендации по организации и выполнению самостоятельной работы

Условия для самостоятельной работы студентов, в частности, для развития навыков самоконтроля, способствующих интенсификации учебного процесса, обеспечиваются наличием и полной доступностью электронных (и бумажных) учебно-методических пособий по основным разделам дисциплины, а так же тестами для самоконтроля по основным разделам дисциплины.

Диагностика компетенций студентов

Текущий контроль усвоения знаний рекомендуется осуществлять в виде проверки отчетов по лабораторным работам, контрольных работ, проведения электронного тестирования.