

УТВЕРЖДАЮ
Зам. директора института кибернетики
по учебной работе

_____ Гайворонский С.А.
«__» _____ 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ

НАПРАВЛЕНИЕ ООП 231000 Программная инженерия

КВАЛИФИКАЦИЯ (СТЕПЕНЬ) бакалавр

БАЗОВЫЙ УЧЕБНЫЙ ПЛАН ПРИЕМА 2015 г.

КУРС 3 СЕМЕСТР 6

КОЛИЧЕСТВО КРЕДИТОВ 6

ПРЕРЕКВИЗИТЫ Б1.В6, Б1.В8, Б1.В13

КОРЕКВИЗИТЫ Б1.В14, Б1.В16

ВИДЫ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ И ВРЕМЕННОЙ РЕСУРС:

Лекции 32 час.

Лабораторная работа 48 час.

Практические занятия 0 час.

АУДИТОРНЫЕ ЗАНЯТИЯ 80 час.

САМОСТОЯТЕЛЬНАЯ РАБОТА 136 час.

ИТОГО 216 час.

ФОРМА ОБУЧЕНИЯ очная

ВИД ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ экзамен

ОБЕСПЕЧИВАЮЩЕЕ ПОДРАЗДЕЛЕНИЕ кафедра ОСУ

ЗАВЕДУЮЩИЙ КАФЕДРОЙ _____ М.А. Иванов

РУКОВОДИТЕЛЬ ООП _____ Е.С. Чердынцев

ПРЕПОДАВАТЕЛЬ _____ А.Ф. Тузовский

2015 г.

1. Цели освоения дисциплины

Цель данной дисциплины – дать обзор современных подходов к разработке архитектуры и проектированию программного обеспечения, изучить и освоить методики выполнения объектно-ориентированного проектирования с использованием языка UML, научить использованию интегрированной системы разработки программного обеспечения Microsoft Visual Studio.

Поставленные цели полностью соответствуют целям (Ц1, Ц2, Ц3) ООП.

Код цели	Формулировка цели
Ц1	Подготовка выпускников к проектно-конструкторской деятельности в области создания и внедрения аппаратных и программных средств объектов профессиональной деятельности в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
Ц2	Подготовка выпускников к проектно-технологической деятельности в области создания компонентов программных комплексов и баз данных, автоматизации технологических процессов с использованием современных инструментальных средств и технологий программирования.
Ц3	Подготовка выпускников к комплексным инженерным исследованиям для решения задач, связанных с разработкой аппаратных и программных средств объектов профессиональной деятельности.

2. Место дисциплины в структуре ООП

Дисциплина «Проектирование и архитектура программных систем» относится к **вариативной части профессионального цикла Б1.В.2** учебных дисциплин ООП.

Для изучения дисциплины «Проектирование и архитектура программных систем» студент должен:

Знать:

- основы объектно-ориентированного подхода к разработке программного обеспечения;
- основы технологии разработки программного обеспечения;
- основы выявления и описания требований заказчика к программному обеспечению.

Уметь:

- ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий;
- программировать на одном из алгоритмических языков;

Владеть:

- основами алгоритмизации и программирования.

Пререквизиты:

Б1.В6 (Программирование), Б1.В8 (Технологии программирования), Б1.В13 (Разработка и анализ требований)

Коррективы:

Б1.В14 (Конструирование программного обеспечения), Б1.В16 (Тестирование программного обеспечения)

3. Результаты освоения дисциплины

Планируемым результатом освоения данной дисциплины является способность ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей (РЗ).

В частности, разрабатывать архитектуры и проекты будущего программного обеспечения в соответствии с техническим заданием и с использованием моделирования и современных методик.

В результате освоения дисциплины студент должен

знать:

- организации процесса проектирования программного обеспечения (ПО) (З.3.1.1);
- методов проектирования структуры ПО (З.3.1.2);
- технологических средств разработки ПО (З.3.1.3).

уметь:

- правильно выбирать архитектурный стиль (У.3.1.1);
- выполнять проектирование подсистем и компонент (У.3.1.2);
- выполнять детальное проектирование программного продукта (У.3.1.3);
- оценивать качество созданного проекта (У.3.1.4).

владеть:

- навыками моделирования программного продукта на языке UML (В.3.1.1);
- навыками использования интегрированной системы Microsoft Visual Studio для моделирования программных систем (В.3.1.2).

В процессе освоения дисциплины у студентов развиваются следующие компетенции:

1. Универсальные (общекультурные)

- владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановки цели и путей ее достижения (ОК-1, ФГОС);
- стремится к саморазвитию, повышению своей квалификации и мастерства (ОК-6, ФГОС);
- имеет навыки работы с компьютером как средством управления информацией (ОК-12, ФГОС);

- способен работать с информацией в глобальных компьютерных сетях (ОК-8, ФГОС).

2. Профессиональные:

Выпускник должен обладать следующими профессиональными компетенциями (ПК):

способен:

- осваивать методики использования программных средств для решения практических задач (ПК-2, ФГОС);
- разрабатывать интерфейсы «человек - ЭВМ» (ПК-3, ФГОС);
- разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования (ПК-5, ФГОС).

4. Структура и содержание дисциплины

4.1. Содержание разделов дисциплины:

Тема № 1. Основные понятия проектирования программного обеспечения
Цели проектирования ПО. Место проектирования ПО в жизненном цикле ПО. Последовательность проектирования ПО.
Тема № 2. Унифицированный язык моделирования UML
Использование моделирования в проектировании ПО. Назначение языка UML. История создания и развития. Основные диаграммы. Недостатки языка UML.
Тема № 3. Анализ требований
Классы анализа. Взаимодействие классов без учета состояний. Взаимодействие классов с учетом состояний. Диаграммы состояний (конечные автоматы).
Тема № 4. Архитектура ПО
Понятие архитектуры ПО. Архитектурные паттерны ПО.
Тема № 5. Проектирование архитектуры
Укрупненное проектирование ПО. Формирование архитектуры программной системы. Оценка качества архитектур ПС.
Тема № 6. Детальное проектирование ПО
Классы проектирования. Уточнение классов анализа. Проектирование взаимодействия классов.
Тема № 7. Паттерны проектирования
Понятия паттерна проектирования. Типы паттернов проектирования. Основные паттерны проектирования.
Тема № 8. Использование компонент при проектировании ПО
Проектирование в конкретных классах и проектирование в интерфейсах.
Тема № 9. Проектирование компонентов ПО.
Принципы проектирования компонентов

4.2. Структура дисциплины по разделам и формам организации обучения

Таблица 1.

Структура модуля (дисциплины) по разделам и формам организации обучения

Название раздела/темы	Аудиторная работа (час)			СРС (час)	Колл, Контр.Р.	Итого
	Лекции	Практ./сем. занятия	Лаб. зан.			
1. Основные понятия проектирования ПО	4		4	10		18
2. Унифицированный язык моделирования UML	4		6	18		28
3. Архитектура ПО	4		6	18		28
4. Основные принципы проектирования ПО	4		6	16	КР-1	26
5. Проектирование архитектуры ПО	4		6	18		28
6. Детальное проектирование ПО	6		6	20		32
7. Шаблоны проектирования	6		8	20		34
8. Оценка качества проектов ПО	4		6	16	КР-2	26
Итого	36	0	48	136		216

4.3 Распределение компетенций по разделам дисциплины

Таблица 2.

Распределение по разделам дисциплины планируемых результатов обучения

№	Формируемые компетенции	Темы дисциплины							
		1	2	3	4	5	6	7	8
1.	З.3.1.1	*							
2.	З.3.1.2		*						
3.	З.3.1.3		*						
4.	У.3.1.1			*	*	*	*	*	
5.	У.3.1.2		*			*			*
6.	У.3.1.3			*			*	*	
7.	У.3.1.4				*			*	
8.	В.3.1.1.		*	*	*	*	*	*	*
9.	В.3.1.2.			*		*	*		

5. Образовательные технологии

Таблица 3.

Методы и формы организации обучения (ФОО)

Методы \ ФОО	Лекц.	Лаб. раб.	Пр. зан./ сем.	Тр* / Мк*	СРС	К. пр.
ИТ-методы		4			8	
Работа в команде						
Case-study						
Игра						
Методы проблемного обучения.	4	4			8	
Обучение на основе опыта						
Опережающая самостоятельная работа	2	4			8	
Проектный метод		4			8	
Поисковый метод		4			8	
Исследовательский метод	4	4			8	
Другие методы						

* - Тренинг, ** - Мастер-класс

6. Организация и учебно-методическое обеспечение самостоятельной работы студентов

6.1. Самостоятельную работу студентов (СРС) делится на текущую и творческую

Текущая СРС :

- работа с лекционным материалом, поиск и обзор литературы и электронных источников информации по индивидуально заданной проблеме курса,
- выполнение домашних заданий, домашних контрольных работ,
- опережающая самостоятельная работа,
- изучение тем, вынесенных на самостоятельную проработку,
- подготовка к лабораторным работам;
- подготовка к контрольной работе;
- подготовка к зачету.

Творческая проблемно-ориентированная самостоятельная работа (ТСР):

- поиск, анализ, структурирование и презентация информации,
- исследовательская работа и участие в научных студенческих конференциях, семинарах и олимпиадах.

6.2. *Содержание самостоятельной работы студентов по дисциплине*

1. Подготовка к следующим лекциям по конспектам, презентациям и пособию.
2. Подготовка к лабораторным работам путем изучения требуемого теоретического материала.
3. Изучение заданного перечня самостоятельно осваиваемых тем

6.3 *Контроль самостоятельной работы*

Оценка результатов самостоятельной работы будет вестись в ходе выполнения контрольных работ и сдачи экзамена.

7. Средства (ФОС) текущей и итоговой оценки качества освоения дисциплины

7.1. Входной контроль (пример вопросов тестовой форме)

1. Что такое программное обеспечение?
2. Жизненный цикл программного обеспечения.
3. Модели разработки программного обеспечения
4. Объектно-ориентированный подход.
5. Модель «водопада» разработки программного обеспечения.

7.3. Рубежный контроль (пример вопросов тестовой форме)

1. Основные диаграммы языка UML.
2. Подсистемы и компоненты.
3. Архитектурные стили.
4. Клиент-серверное программное обеспечение.
5. Диаграмма классов.
6. Сервисно-ориентированное ПО.
7. Принцип открытости закрытости ПО.

Пример вопросов для экзамена

1. Объектно-ориентированный анализ (ООА). Модели ООА.
2. Классы анализа. Их выявление и описание.
3. Построение статической модели анализа.
4. Динамической модели без учета состояния. Порядок построения и описания.
5. Конечные автоматы.
6. Динамической модели с учетом состояния. Порядок построения и описания.
7. Объектно-ориентированное проектирование (ООП).
8. Проектирование классов программного обеспечения.
9. Проектирование связей между классами программного обеспечения.
10. Уточнение отношений между классами, выявленными на этапе анализа.

- 11.Выделение подсистем на этапе проектирования.
- 12.Шаблоны архитектуры программного обеспечения.
- 13.Интерфейсы и компоненты.

8. Рейтинг качества освоения дисциплины

Дисциплина	Проектирование и архитектура программных систем	Число недель	18
Институт	Кибернетики	Кол-во кредитов	6
Кафедра	ОСУ	Лекции, час	32
Семестр	6	Практич. занятия, час	
Группы	3 курс	Лаб. работы, час.	48
Преподаватель	Тузовский А.Ф., профессор	Всего аудит. работы, час	80
		Самост. работа, час	136
		ВСЕГО, час	216

Рейтинг-план освоения дисциплины в течение семестра

Недели	Текущий контроль													Итого
	Теоретический материал				Практическая деятельность									
	Название модуля	Темы лекций	Контрол-ир. ма-тер.	Бал-лы	Название лаборатор-ных работ	Баллы	Темы практиче-ских заня-тий	Бал-лы	Рубежные контрольные рабо-ты	Бал-лы	Проблемно-ориентирован-ные задания	Бал-лы		
1	Основные понятия проектирования ПО	Основные понятия проектирования ПО		1	Создание UML-диаграмм	2							3	
2	Основные понятия проектирования ПО	Основные понятия проектирования ПО		1	Создание UML-диаграмм	3							4	
3	Унифицированный язык моделирования UML	Унифицированный язык моделирования UML		1	Проектирование классов	2							3	

Недели	Текущий контроль													
	Теоретический материал				Практическая деятельность									Итого
	Название модуля	Темы лекций	Контрол-ир. ма-тер.	Бал-лы	Название лаборатор-ных работ	Баллы	Темы практиче-ских заня-тий	Бал-лы	Рубежные контроль-ные рабо-ты	Бал-лы	Проблемно-ориентирован-ные задания	Бал-лы		
4	Унифици-рованный язык мо-делирова-ния UML	Унифицированный язык моделирова-ния UML		1	Проектирование классов	3							4	
5	Архитек-тура ПО	Архитектура ПО		1	Проектирование вза-имдействия классов	3							4	
6	Архитек-тура ПО	Архитектура ПО		1	Проектирование вза-имдействия классов	3							4	
7	Основные принципы проекти-рования ПО	Основные принци-пы проектирования ПО		1	Проектирование вза-имдействия классов	3							4	
8	Основные принципы проекти-рования ПО	Основные принци-пы проектирования ПО		1	Проектирование вза-имдействия классов	3							4	
9		Коференц-неделя												
Всего по контрольной точке (аттестации) № 1													30	
10	Укруп-ненное проекти-рование ПО	Укрупненное про-ектирование ПО		1	Формирование под-систем	2							3	

Недели	Текущий контроль													
	Теоретический материал				Практическая деятельность									Итого
	Название модуля	Темы лекций	Контрол-ир. ма-тер.	Бал-лы	Название лаборатор-ных работ	Баллы	Темы практиче-ских заня-тий	Бал-лы	Рубежные контрольные рабо-ты	Бал-лы	Проблемно-ориентирован-ные задания	Бал-лы		
11	Укруп-ненное проекти-рование ПО	Укрупненное про-ектирование ПО		1	Формирование под-систем	3							4	
12	Детальное проекти-рование ПО	Детальное проекти-рование ПО		1	Формирование под-систем	3							4	
13	Детальное проекти-рование ПО	Детальное проекти-рование ПО		1	Шаблоны проектиро-вания	3							4	
14	Детальное проекти-рование ПО	Детальное проекти-рование ПО		1	Шаблоны проектиро-вания	3							4	
15	Шаблоны проекти-рования	Шаблоны проекти-рования		1	Шаблоны проектиро-вания	3							4	
16	Шаблоны проекти-рования	Шаблоны проекти-рования		1	Шаблоны проектиро-вания	3							4	
17	Оценка качества проектов ПО	Оценка качества проектов ПО		1	Оценка качества про-екта ПО	2							3	
18		Коференц-неделя												
Всего по контрольной точке (аттестации) № 2													30	
Итоговая текущая аттестация													60	

Недели	Текущий контроль													Итого
	Теоретический материал				Практическая деятельность									
	Название модуля	Темы лекций	Контрол-лир. матер.	Бал-лы	Название лаборатор-ных работ	Баллы	Темы практиче-ских заня-тий	Бал-лы	Рубежные контроль-ные рабо-ты	Бал-лы	Проблемно-ориентирован-ные задания	Бал-лы		
Экзамен (зачет)													40	
Итого баллов по дисциплине													100	

9. Учебно-методическое и информационное обеспечение дисциплины

1. Макконнелл, Стив Совершенный код : практическое руководство по разработке программного обеспечения : пер. с англ. / С. Макконнелл. — Москва: Русская редакция, 2013. — 869 с.
2. Брауде, Эрик Дж. Технология разработки программного обеспечения / Э. Д. Брауде. — СПб.: Питер, 2004. — 655 с.
3. Орлов, Сергей Александрович Технологии разработки программного обеспечения. Современный курс по программной инженерии : учебник / С. А. Орлов, Б. Я. Цилькер. — 4-е изд.. — СПб.: Питер, 2012. — 608 с.
4. Рамбо, Дж. UML 2.0. Объектно-ориентированное моделирование и разработка : пер. с англ. / Дж. Рамбо, М. Блаха. — 2-е изд. — СПб.: Питер Пресс, 2007. — 544 с.
5. Буч Г. Язык UML. Руководство пользователя
http://e.lanbook.com/books/element.php?pl1_id=1246

Дополнительная литература:

1. Гагарина Л.Г. Технология разработки программного обеспечения: учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. — Москва: Форум Инфра-М, 2013. — 400 с.
2. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. — 2-е изд. — СПб.: Питер, 2006.
3. Круз Р.Л. Структуры данных и проектирование программ: пер. с англ. / Р. Л. Круз. — Москва: БИНОМ. Лаборатория знаний, 2008. — 765 с.
4. Мацяшек Л.А. Практическая программная инженерия на основе учебного примера : пер. с англ. / Л. А. Мацяшек, Б. Л. Лионг. — Москва: БИНОМ. Лаборатория знаний, 2009. — 956 с.
5. Гома Х. — UML. Проектирование систем реального времени, параллельных и распределенных приложений <http://e.lanbook.com/view/book/1232>

Программное обеспечение и *Internet*-ресурсы:

- Visual Studio Express Edition
- Учебно-методические материалы, находящиеся в локальной сети кафедры ОСУ.

10. Материально-техническое обеспечение модуля (дисциплины)

Компьютерный класс – 10 компьютеров Pentium IV(MB S-478 Bayfield D865GBFL i865G 800 MHz, Celeron 2.4GHz, 2 Dimm 256 Mb, HDD 40 Gb)

Программа составлена на основе Стандарта ООП ТПУ в соответствии с требованиями ФГОС по направлению и профилю подготовки 231000 «Программная инженерия».

Программа одобрена на заседании кафедры ОСУ
(протокол № _____ от «___» _____ 20__ г.).

Автор д.т.н., профессор каф. ОСУ _____ А.Ф. Тузовский

Рецензент(ы) _____