


Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)

УТВЕРЖДАЮ
Проректор
по учебно-методической работе

А.А. Панфилов
« 09 » _____ 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ ПРОЕКТИРОВАНИЯ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ СИСТЕМ УПРАВЛЕНИЯ»

Направление подготовки *27.04.04 Управление в технических системах*

Профиль подготовки: *Управление и информатика в технических системах*

Уровень высшего образования: *магистратура*

Форма обучения: *очная*

| Семестр | Трудоем- кость зач, ед,час. | Лек- ций, час. | Практик. занятий, час. | Лаборат. работ, час. | СРС, час. | Форма промежуточного контроля (экз./зачет) |
|--------------|-----------------------------------|----------------------|------------------------------|----------------------------|--------------|---|
| 2 | 4/144 | 18 | 18 | 18 | 90 | зачет |
| Итого | 4/144 | 18 | 18 | 18 | 90 | зачет |

Владимир, 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины (модуля) «Технологии проектирования программного обеспечения систем управления» является: формирование системы понятий, знаний, умений и навыков в области современных технологий разработки программного обеспечения для систем управления.

Задачи дисциплины:

- изучение теоретических основ разработки ПО
- знакомство с программными средствами для автоматизации разработки ПО
- знакомство с современными языками и технологиями программирования

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

В структуре ОПОП ВО по направлению **27.04.04 Управление в технических системах** дисциплина «Технологии проектирования программного обеспечения систем управления» является дисциплиной по выбору вариативной части программы магистратуры.

По «входу» дисциплина «Технологии проектирования программного обеспечения систем управления» основывается на изучении дисциплин «Компьютерные технологии управления в технических системах», «Автоматизированное проектирование средств и систем управления».

Дисциплина «Технологии проектирования программного обеспечения систем управления» является предшествующей для следующих дисциплин «Проектирование систем автоматического управления», «Современные интернет-технологии в информационно-управляющих системах».

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Дисциплина «Технологии проектирования программного обеспечения систем управления» участвует в формировании следующих компетенций:

- способность применять современные методы разработки технического, информационного и алгоритмического обеспечения систем автоматизации и управления (ПК-3);

- способность применять современный инструментарий проектирования программно-аппаратных средств для решения задач автоматизации и управления (ПК-6);
- способностью выбирать методы и разрабатывать алгоритмы решения задач управления в технических системах (ПК-8);

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования.

1) Знать:

- стандарты в области разработки ПО (ПК-6, ПК-8)
- основные этапы жизненного цикла разработки ПО и выполняемые на них действия (ПК-6)
- основные операторы и структуры данных языка С# (ПК-3)
- основные принципы структурного и объектно-ориентированного программирования (ПК-3, ПК-8)
- основные классы библиотеки .NET Framework (ПК-3, ПК-8)

2) Уметь:

- использовать современные подходы к разработке ПО (ПК-6, ПК-8)
- проектировать программные средства с использованием моделей (ПК-6)
- создавать программы для решения задач предметной области при помощи языка С# (ПК-3)
- выполнять тестирование и отладку программ на языке С# (ПК-3)

3) Владеть

- современными программными средствами для автоматизации разработки ПО (ПК-6)
- современными интегрированными средами разработки приложений (ПК-8)

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 часа.

| № ПП | Раздел (тема) дисциплина | Семестр | Неделя семестра | Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах) | | | | | | | Объем уч работы с применением интерактивных методов (в час/%) | Формы текущего контроля успеваемости. Форма промежуточной аттестации | |
|---------|---------------------------|---------|-----------------|--|----------|----------------------|---------------------|--------------------|-----|-------|---|--|-------|
| | | | | Лекции | Семинары | Практические занятия | Лабораторные работы | Контрольные работы | СРС | КП/КР | | | |
| 1 | Процесс разработки ПО | 2 | 1-6 | 6 | | 6 | | | | 30 | | 6/50% | 1 р-к |
| 2 | Проектирование ПО | 2 | 6-12 | 4 | | 4 | 10 | | | 30 | | 4/22% | 2 р-к |
| 3 | Разработка ПО на языке С# | 2 | 12-18 | 8 | | 8 | 8 | | | 30 | | 8/33% | 3 р-к |
| Всего | | | | 18 | | 18 | 18 | | | 90 | | 18/33% | Зачет |

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При проведении занятий по дисциплине «Технологии проектирования программного обеспечения систем управления» предполагается использовать следующие образовательные технологии: при проведении лекционных и практических занятий использование мультимедийных технологий, основанных на презентациях в среде Power Point, использование демоверсий примеров применения пакетов прикладных программ.

При проведении практических и лабораторных занятий комбинирование различных по сложности заданий, предполагающих как решение типовых задач, так и задач по индивидуальным заданиям, требующих самостоятельного решения, интерактивное обсуждение результатов по индивидуальным заданиям. При подготовке к выполнению индивидуальных заданий студенты изучают литературу по соответствующей проблемной области, проводят поиск необходимых источников в Интернете.

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Текущий контроль успеваемости и качества подготовки студентов проводится с учетом посещения всех видов занятий, выполнения заданий во время практических и ла-

бораторных занятий, заданий для самостоятельной работы и результатам рейтингового контроля.

Промежуточная аттестация по данной дисциплине проводится в виде зачета.

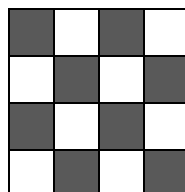
КОНТРОЛЬНЫЕ ВОПРОСЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Перечислите основные этапы развития технологий разработки ПО.
2. Что нового в разработке ПО возникло на этапе структурного программирования?
3. Что такое процедурное программирование?
4. Что такое модульное программирование?
5. В чем заключаются основные идеи объектного подхода к программированию?
6. Что такое стандарт? Стандартизация?
7. Зачем необходима стандартизация процесса разработки ПО?
8. Приведите примеры стандартов в области ПО.
9. Опишите стандарт ISO 12 207.
10. Что такое ЕСПД?
11. Что такое жизненный цикл ПО?
12. Что такое модель и методология жизненного цикла?
13. Какие модели жизненного цикла ПО Вы знаете?
14. Охарактеризуйте каскадную модель жизненного цикла разработки ПО.
15. Охарактеризуйте итеративную модель жизненного цикла разработки ПО.
16. Охарактеризуйте спиральную модель жизненного цикла разработки ПО.
17. Каковы, на Ваш взгляд, основные преимущества и недостатки различных моделей жизненного цикла?
18. Перечислите основные стадии создания ПО в соответствии с ГОСТ 34.601-90.
19. Какие процессы жизненного цикла разработки ПО регламентируются стандартом ISO 12 207?
20. Что такое Rational Objectory Process?
21. Каковы основные особенности гибкой методологии разработки ПО?
22. Перечислите основные принципы Agile
23. Каковы основные недостатки гибкой методологии разработки ПО?
24. Что такое экстремальное программирование?
25. В чем заключаются структурный и объектный подходы к проектированию ПО?
26. Какие модели строятся при проектировании ПО в рамках структурного подхода? Объектного подхода?
27. Какими способами можно описывать спецификации процессов?
28. С какой целью используются диаграммы переходов состояний?
29. В чем смысл принципа декомпозиции функциональных диаграмм?
30. С какой целью используются диаграммы потоков данных?
31. Что такое UML?
32. Из каких структурных компонентов состоит UML?
33. Какие задачи можно решать при помощи диаграммы прецедентов?
34. Какие задачи можно решать при помощи диаграммы классов?
35. Что такое CASE-средство и зачем они нужны?
36. На каких этапах разработки ПО чаще всего используются CASE-средства?
37. Перечислите основные типы CASE-средств.
38. Приведите примеры CASE-средств, представленных на российском рынке.
39. Для решения каких задач была разработана платформа Microsoft.NET Framework?
40. Каковы отличия различных версий платформы Microsoft.NET Framework?
41. Что такое CLR?
42. В чем заключается механизм JIT?

43. Какие языки программирования для платформы Microsoft.NET Framework Вы знаете?
44. Почему язык C# считается основным языком разработки .NET приложений?
45. Из каких основных элементов состоит язык C#?
46. Что такое CLS?
47. Каковы структура системы типов данных языка C#?
48. Что такое простые и составные типы?
49. Какие числовые типы C# Вы знаете?
50. Какие встроенные методы для работы с числами имеются в .NET?
51. Как описать переменную в C#?
52. Что такое операции? Какие операции C# Вы знаете?
53. Как вывести на экран значение переменной?
54. Как ввести с клавиатуры значение строковой переменной? Числовой переменной?
55. Какие операторы языка C# вы знаете?
56. Как записывается оператор присваивания?
57. Почему оператор безусловного перехода не рекомендуется использовать?
58. Как записывается и работает оператор ветвления?
59. Какие циклы существуют в C#? В чем разница между ними?
60. В каких случаях используется каждый из операторов цикла?
61. Какие операторы можно использовать для досрочного прекращения выполнения цикла?
62. Как записывается и работает оператор выбора?
63. Что такое исключение?
64. Зачем необходимо обрабатывать исключения?
65. Как осуществляется обработка исключений в C#?
66. В чем причина появления парадигмы объектно-ориентированного программирования?
67. Что такое класс и объект?
68. Поясните принципы инкапсуляции, наследования, полиморфизма.
69. Как описывается класс в C#?
70. Что такое модификаторы доступа?
71. Что такое свойства? Как описать свойство в C#?
72. Зачем нужны конструкторы и деструкторы?
73. Как описываются методы в C#?
74. Как достигается полиморфизм методов?
75. Как осуществляется наследование в C#?

КОНТРОЛЬНЫЕ ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Задача 1. Дана шахматная доска размером $n \times n$ клеток. Верхняя левая клетка доски черная и имеет номер (1, 1). Например, для $n=4$ шахматная таблица выглядит следующим образом:



- 1) для заданного значения n определить количество черных ячеек шахматной доски;
- 2) по номеру ячейки (k, m) определить ее цвет;
- 3) определить, являются ли ячейки с номерами (k_1, m_1) и (k_2, m_2) одного цвета;

- 4) определить, находится ли фигура, стоящая в ячейке с номером (k_1, m_1) , под ударом второй фигуры, стоящей в ячейке с номером (k_2, m_2) , при условии, что ход второй фигуры и ей является: а) пешка; б) слон; в) ладья; г) ферзь; е) конь.

Задача 2. Задана дата в формате <день>.<месяц>.<год>. Определить:

- 1) сколько дней прошло с начала года;
- 2) сколько дней осталось до конца года;
- 3) дату предыдущего дня;
- 4) дату следующего дня.

Задача 3. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n -ную степень, равна самому числу. Например, $153=1^3+5^3+3^3$. Найти все трехзначные числа Армстронга.

Задача 4. Стороны прямоугольника заданы натуральными числами n и m . Найти количество квадратов (стороны которых выражены натуральными числами), на которые можно разрезать данный прямоугольник, если от него каждый раз отрезать квадрат:

- 1) наименьшей площади;
- 2) наибольшей площади

Задача 5. В одномерном массиве, элементы которого – целые числа, произвести следующие действия:

1. Удалить из массива все четные числа.
2. Вставить новый элемент после всех элементов, которые заканчиваются на данную цифру.
3. Удалить из массива повторяющиеся элементы, оставив только их первые вхождения.
4. Вставить новый элемент между всеми парами элементов, имеющими разные знаки.
5. Уплотнить массив, удалив из него все нулевые значения.

ВОПРОСЫ К ЗАЧЕТУ

1. Большие программные проекты. Причины успехов и неудач программных проектов.
2. История методов разработки ПО.
3. Роль стандартов в разработке ПО.
4. Обзор стандартов группы ГОСТ 19 и ГОСТ 34.
5. Международный стандарт ISO 12207.
6. Жизненный цикл разработки ПО.
7. Современные подходы к организации жизненного цикла разработки ПО.
8. Основные принципы структурного подхода к разработке ПО.
9. Диаграммы потоков данных. Диаграммы сущность-связь.
10. Диаграммы переходов состояний. Функциональные диаграммы.
11. Основные принципы объектно-ориентированного подхода к разработке ПО.
12. Язык UML.
13. CASE-технологии. Современные CASE-средства.
14. Методология экстремального программирования.
15. Платформа MS .NET Framework
16. Структура программы на языке C#. Переменные. Области видимости. Операторы.
17. Простые типы данных в C#.
18. Операторы ветвления и выбора.
19. Операторы циклов.
20. Работа с массивами в C#.
21. Обработка строк в C#.
22. ООП. Классы. Объекты. Методы. Свойства.. События.
23. ООП. Инкапсуляция, наследование, полиморфизм.
24. Описание класса в C#.
25. Описание методов в C#. Конструкторы и деструкторы.

26. Перегрузка методов в C#.
27. Поля и свойства в C#.

КОНТРОЛЬНЫЕ УЧЕБНЫЕ ЗАДАНИЯ

ДЛЯ ПРОВЕДЕНИЯ РЕЙТИНГ-КОНТРОЛЯ № 1 (образец заданий)

1. Что нового в разработке ПО возникло на этапе структурного программирования?
2. В чем состоят основные преимущества компонентного подхода к программированию?
3. Зачем необходима стандартизация процесса разработки ПО?
4. Как в экстремальном программировании удается решить проблему пирамиды стоимости ошибок?
5. Можно ли изолированно применять отдельные практики экстремального программирования?

КОНТРОЛЬНЫЕ УЧЕБНЫЕ ЗАДАНИЯ

ДЛЯ ПРОВЕДЕНИЯ РЕЙТИНГ-КОНТРОЛЯ № 2 (образец заданий)

1. Какие модели строятся при проектировании ПО в рамках структурного подхода? Объектного подхода?
2. Из каких структурных компонентов состоит UML?
3. Какие задачи можно решать при помощи диаграммы прецедентов?
4. Какие задачи можно решать при помощи диаграммы классов?
5. Что такое CASE-средство и зачем они нужны?

КОНТРОЛЬНЫЕ УЧЕБНЫЕ ЗАДАНИЯ

ДЛЯ ПРОВЕДЕНИЯ РЕЙТИНГ-КОНТРОЛЯ № 3 (образец заданий)

1. Для решения каких задач была разработана платформа Microsoft.NET Framework?
2. Почему язык C# считается основным языком разработки .NET приложений?
3. Почему оператор безусловного перехода не рекомендуется использовать?
4. Зачем необходимо обрабатывать исключения?
5. Как описывается класс в C#?

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература

1. Основы Microsoft .NET Framework и языка программирования С#: [Электронный ресурс] / Суханов М.В. - Архангельск : ИД САФУ, 2014. - <http://www.studentlibrary.ru/>
2. Объектно-ориентированное программирование в С++: лекции и упражнения [Электронный ресурс] : Учебное пособие для вузов / Ашарина И.В. - М. : Горячая линия - Телеком, 2012. - <http://www.studentlibrary.ru/>
3. NI Measurement Studio: практика разработки систем измерения и управления на С# [Электронный ресурс] / Магда Ю.С. - М. : ДМК Пресс, 2013. - <http://www.studentlibrary.ru/>

б) дополнительная литература

1. .NET Framework 2.0. Секреты создания Windows-приложений [Электронный ресурс] / С.С. Байдачный - М. : СОЛОН-ПРЕСС, 2008. - <http://www.studentlibrary.ru/>
2. Прикладное программирование на С/С++: с нуля до мультимедийных и сетевых приложений [Электронный ресурс] / В.Б. Иванов - М. : СОЛОН-ПРЕСС, 2008. - <http://www.studentlibrary.ru/>
3. Разработка приложений Java EE 6 в NetBeans 7 [Электронный ресурс] / Дэвид Хеффельфингер ; Пер. с англ.: Карышев Е.Н. - М. : ДМК Пресс, 2013. - <http://www.studentlibrary.ru/>
4. Стандартизация и разработка программных систем [Электронный ресурс] / учеб. пособие / В.Н. Гусятников, А.И. Безруков. - М.: Финансы и статистика, 2010. - http://www.studentlibrary.ru
5. Проектирование автоматизированных систем производства [Электронный ресурс] : Учеб. пособие / В.Л. Конюх. - М. : Абрис, 2012. - <http://www.studentlibrary.ru/>
6. Проектирование автоматизированных систем обработки информации и управления (АСОИУ) [Электронный ресурс] : учебник / Я.А. Хетагуров. - М. : БИНОМ, 2015. - (Учебник для высшей школы). - <http://www.studentlibrary.ru/>
7. Технология проектирования автоматизированных систем обработки информации и управления [Электронный ресурс] : Учебное пособие для вузов / Рудинский И.Д. - М. : Горячая линия - Телеком, 2011. - <http://www.studentlibrary.ru/>

в) периодические издания

1. Журнал «MSDN Magazine». М.: Русская Редакция. 2011-2015

г) Интернет-ресурсы

1. intuit.ru
2. [.ru.wikipedia.org](http://ru.wikipedia.org)
3. msdn.microsoft.com/ru-ru

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Практические занятия проводятся в аудитории, обеспеченной мультимедийной аппаратурой, позволяющей использовать различные варианты демонстрации изучаемого материала.

Студенты имеют возможность доступа к локальной сети кафедры и сети университета.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению «Управление в технических системах»

Рабочую программу составил:



А.В.Шутов
к.ф.-м.н., доцент

Рецензент

Нач.лаборатории
ЗАО «Автоматика плюс» к.т.н.



В.М.Дерябин

Программа рассмотрена и одобрена на заседании кафедры УИТЭС

Протокол № 1/1 от 6.02.15 года

Заведующий кафедрой



А.Б.Градусов

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления «Управление в технических системах»

Протокол № 2 от 6.02.15 года

Председатель комиссии



А.Б.Градусов