МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Технология программирования

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки (специальность): **09.03.01 Информатика и вычислительная техника**
Направленность (профиль): **Автоматизированное управление бизнес-процессами и финансами**
Форма обучения: **очная**
Факультет: **ФВС, Факультет вычислительных систем**
Кафедра: **ЭМИС, Кафедра экономической математики, информатики и статистики**
Курс: **3**
Семестр: **6**
Учебный план набора 2015 года

Распределение рабочего времени

№	Виды учебной деятельности	6 семестр	Всего	Единицы
1	Лекции	34	34	часов
2	Практические занятия	34	34	часов
3	Лабораторные занятия	34	34	часов
4	Всего аудиторных занятий	102	102	часов
5	Самостоятельная работа	114	114	часов
6	Всего (без экзамена)	216	216	часов
7	Подготовка и сдача экзамена	36	36	часов
8	Общая трудоемкость	252	252	часов
		7.0	7.0	З.Е

Экзамен: 6 семестр

Томск 2016

Согласована на портале № 4406

Рассмотрена и одобрена на заседании кафедры
протокол № 2 от «14» 10 2016 г.

ЛИСТ СОГЛАСОВАНИЙ

Рабочая программа составлена с учетом требований Федерального Государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника, утвержденного 2016-01-12 года, рассмотрена и утверждена на заседании кафедры «___» _____ 20__ года, протокол №_____.

Разработчики:

доцент каф. ЭМИС _____ Зариковская Н. В.

Заведующий обеспечивающей каф. ЭМИС _____ Боровской И. Г.

Рабочая программа согласована с факультетом, профилирующей и выпускающей кафедрами направления подготовки (специальности).

Декан ФВС _____ Козлова Л. А.

Заведующий выпускающей каф. ЭМИС _____ Боровской И. Г.

Эксперты:

доцент кафедра ЭМИС _____ Шельмина Е. А.

Согласована на портале № 4406

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Целями освоения учебной дисциплины «Технология программирования» являются

овладение основными принципами, базовыми концепциями технологий программирования, основными этапами и принципами создания программного продукта;

интеллектуальное развитие, формирование качеств личности, необходимых человеку для полноценной жизни в современном обществе: ясность и точность мысли, интуиция, элементы алгоритмической культуры, пространственных представлений, способность к преодолению трудностей, продолжения образования;

формирование представлений об основных принципах, базовых концепциях технологий программирования, основных этапах и принципах создания программного продукта;

воспитание культуры личности, отношения к точным наукам как к части общечеловеческой культуры, понимание их значимости для научно-технического прогресса.

1.2. Задачи дисциплины

– Задачами дисциплины «Технология программирования» являются освоение:

– основных принципов и базовых концепций технологий программирования, основных этапов и принципов создания программного продукта,

– технологии разработки алгоритмов и программ;

– освоение методов отладки и решения задач на ЭВМ в различных режимах.

2. Место дисциплины в структуре ОПОП

Дисциплина «Технология программирования» (Б1.В.ОД.7) относится к блоку 1 (вариативная часть).

Предшествующими дисциплинами, формирующими начальные знания, являются следующие дисциплины: Объектно-ориентированное программирование, Программирование.

Последующими дисциплинами являются: Управление проектами.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-1 Способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.;

– ОПК-2 Способность осваивать методики использования программных средств для решения практических задач.;

В результате изучения дисциплины студент должен:

– **знать** основные принципы, базовые концепции технологий программирования, основные этапы и принципы создания программного продукта, проблемы сложности, основы проектирования с учетом изменений, обработка исключений, технологию разработки алгоритмов и программ; методы отладки и решения задач на ЭВМ в различных режимах; различия между спецификацией и реализацией, основные стандарты в области инфокоммуникационных систем и технологий, в том числе стандарты единой системы программной документации (ЕСПД);

– **уметь** ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы; использовать прикладные системы программирования; разрабатывать основные программные документы;

– **владеть** методами навыками владения одной из технологий программирования; навыками разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 7.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры
		6 семестр

Согласована на портале № 4406

Аудиторные занятия (всего)	102	102
Лекции	34	34
Практические занятия	34	34
Лабораторные занятия	34	34
Самостоятельная работа (всего)	114	114
Оформление отчетов по лабораторным работам	42	42
Проработка лекционного материала	34	34
Подготовка к практическим занятиям, семинарам	38	38
Всего (без экзамена)	216	216
Подготовка и сдача экзамена	36	36
Общая трудоемкость час	252	252
Зачетные Единицы Трудоемкости	7.0	7.0

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

№	Названия разделов дисциплины	Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа	Всего часов (без экзамена)	Формируемые компетенции
1	Вводная часть	2	0	0	4	6	ОПК-1, ОПК-2
2	Организация процесса проектирования программного обеспечения	4	4	0	10	18	ОПК-1, ОПК-2
3	Методы проектирования программного обеспечения	4	4	0	10	18	ОПК-1, ОПК-2
4	Парадигмы программирования	4	0	22	26	52	ОПК-1, ОПК-2
5	Технология создания программного кода	4	4	8	20	36	ОПК-1, ОПК-2
6	Технологии коллективной разработки программного обеспечения	4	6	0	10	20	ОПК-1, ОПК-2
7	Технологические средства разработки программного обеспечения	4	4	4	14	26	ОПК-1, ОПК-2
8	Методы отладки и тестирования программ	4	6	0	10	20	ОПК-1, ОПК-2
9	Документирование и оценка качества программных продуктов	4	6	0	10	20	ОПК-1, ОПК-2

Согласована на портале № 4406

	Итого	34	34	34	114	216	

5.2. Содержание разделов дисциплины (по лекциям)

Содержание разделов дисциплин (по лекциям) приведено в таблице 5.2.

Таблица 5.2 - Содержание разделов дисциплин (по лекциям)

Названия разделов	Содержание разделов дисциплины по лекциям	Трудоемкость, ч	Формируемые компетенции
6 семестр			
1 Вводная часть	Введение в технологию программирования, программную инженерию. Понятие программного средства. Технология программирования и основные этапы ее развития. Проблемы разработки сложных программных систем. Жизненный цикл программного средства. Архитектура ПО. Системный анализ при создании ПС.	2	ОПК-1, ОПК-2
	Итого	2	
2 Организация процесса проектирования программного обеспечения	Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков	4	ОПК-1, ОПК-2
	Итого	4	
3 Методы проектирования программного обеспечения	Использование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных. Жизненный цикл программного средства.	4	ОПК-1, ОПК-2
	Итого	4	
4 Парадигмы программирования	Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная и т.д. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программировании (инкапсуляция, наследование и полиморфизм); классы	4	ОПК-1, ОПК-2

Согласована на портале № 4406

	и объекты; интерфейсы и реализация.		
	Итого	4	
5 Технология создания программного кода	Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; примеры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглушки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. Технологии распределенных вычислений: RPC, RMI, Corba, DCOM.	4	ОПК-1, ОПК-2
	Итого	4	
6 Технологии коллективной разработки программного обеспечения	Обзор и классификация средств поддержки коллективной разработки ПО. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабочего процесса. Сценарии выполнения работ. Применение систем управления документами. CASE-технологии.	4	ОПК-1, ОПК-2
	Итого	4	
7 Технологические средства разработки программного обеспечения	Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. Отладчики. CASE-технология. UML-диаграммы.	4	ОПК-1, ОПК-2
	Итого	4	
8 Методы отладки и тестирования программ	Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Регрессионное тестирование. Тестирование "черного ящика". Разработка тестов.	4	ОПК-1, ОПК-2
	Итого	4	
9 Документирование и оценка качества программных продуктов	Документация, создаваемая в процессе разработки программных средств. ЕСПД. Пользовательская документация программных средств. Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики	4	ОПК-1, ОПК-2

Согласована на портале № 4406

	качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств.		
	Итого	4	
Итого за семестр		34	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представ-лены в таблице 5.3.

Таблица 5.3 - Разделы дисциплины и междисциплинарные связи

№	Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин								
		1	2	3	4	5	6	7	8	9
Предшествующие дисциплины										
1	Объектно-ориентированное программирование							+	+	
2	Программирование				+	+				
Последующие дисциплины										
1	Управление проектами	+	+				+			+

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4

Таблица 5. 4 – Соответствие компетенций и видов занятий, формируемых при изучении дисциплины

Компетенции	Виды занятий				Формы контроля
	Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа	
ОПК-1	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Коллоквиум, Отчет по лабораторной работе, Опрос на занятиях, Реферат

Согласована на портале № 4406

ОПК-2	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Коллоквиум, Отчет по лабораторной работе, Опрос на занятиях, Реферат

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП

7. Лабораторный практикум

Содержание лабораторных работ приведено в таблице 7.1.

Таблица 7. 1 – Содержание лабораторных работ

Названия разделов	Содержание лабораторных работ	Трудоемкость, ч	Формируемые компетенции
6 семестр			
4 Парадигмы программирования	Инструментальная среда разработки	2	ОПК-1, ОПК-2
	Классы. Создание класса. Сбор мусора	4	
	Средства отладки и обработка ошибок	4	
	Добавление компонентов и реализация членов	4	
	Применение производных классов	4	
	Подключение к базе данных	4	
	Итого	22	
5 Технология создания программного кода	Создание нестандартных элементов управления	4	ОПК-1, ОПК-2
	Создание пользовательского интерфейса, меню	4	
	Итого	8	
7 Технологические средства разработки программного обеспечения	Технологические средства разработки	4	ОПК-1, ОПК-2
	Итого	4	
Итого за семестр		34	

8. Практические занятия

Содержание практических работ приведено в таблице 8.1.

Таблица 8. 1 – Содержание практических работ

Названия разделов	Содержание практических занятий	Трудоемкость, ч	Формируемые компетенции
6 семестр			
2 Организация процесса	Модели разработки: каскадная, с	4	ОПК-1,

Согласована на портале № 4406

проектирования программного обеспечения	промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков		ОПК-2
	Итого	4	
3 Методы проектирования программного обеспечения	Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных. Жизненный цикл программного средства.	4	ОПК-1, ОПК-2
	Итого	4	
5 Технология создания программного кода	Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; многооконные интерфейсы; примеры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглушки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных. Технологии распределенных вычислений: RPC, RMI, Corba, DCOM.	4	ОПК-1, ОПК-2
	Итого	4	
6 Технологии коллективной разработки программного обеспечения	Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабочего процесса. Применение систем управления документами. CASE-технологии.	6	ОПК-1, ОПК-2
	Итого	6	
7 Технологические средства разработки программного обеспечения	Инструментальная среда разработки. Библиотека VCL. Отладчики. CASE-технология. UML-диаграммы.	4	ОПК-1, ОПК-2
	Итого	4	
8 Методы отладки и тестирования программ	Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Регрессионное тестирование. Тестирование "черного ящика". Разработка тестов.	6	ОПК-1, ОПК-2
	Итого	6	
9 Документирование и оценка качества программных продуктов	Документация, создаваемая в процессе разработки программных средств.	6	ОПК-1, ОПК-2

Согласована на портале № 4406

	ЕСПД. Пользовательская документация программных средств. Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства.		
	Итого	6	
Итого за семестр		34	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 - Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость ч	Формируемые компетенции	Формы контроля
6 семестр				
1 Вводная часть	Проработка лекционного материала	4	ОПК-1, ОПК-2	Опрос на занятиях
	Итого	4		
2 Организация процесса проектирования программного обеспечения	Подготовка к практическим занятиям, семинарам	6	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по индивидуальному заданию
	Проработка лекционного материала	4		
	Итого	10		
3 Методы проектирования программного обеспечения	Подготовка к практическим занятиям, семинарам	6	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по индивидуальному заданию
	Проработка лекционного материала	4		
	Итого	10		
4 Парадигмы программирования	Проработка лекционного материала	2	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по лабораторной работе
	Оформление отчетов по лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		

Согласована на портале № 4406

	Оформление отчетов по лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Итого	26		
5 Технология создания программного кода	Подготовка к практическим занятиям, семинарам	4	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Отчет по лабораторной работе
	Проработка лекционного материала	4		
	Оформление отчетов по лабораторным работам	6		
	Оформление отчетов по лабораторным работам	6		
	Итого	20		
6 Технологии коллективной разработки программного обеспечения	Подготовка к практическим занятиям, семинарам	6	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по индивидуальному заданию
	Проработка лекционного материала	4		
	Итого	10		
7 Технологические средства разработки программного обеспечения	Подготовка к практическим занятиям, семинарам	4	ОПК-1, ОПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Отчет по лабораторной работе
	Проработка лекционного материала	4		
	Оформление отчетов по лабораторным работам	6		
	Итого	14		
8 Методы отладки и тестирования программ	Подготовка к практическим занятиям, семинарам	6	ОПК-1, ОПК-2	Контрольная работа, Опрос на занятиях, Отчет по индивидуальному заданию
	Проработка лекционного материала	4		
	Итого	10		
9 Документирование и оценка качества программных продуктов	Подготовка к практическим занятиям, семинарам	6	ОПК-1, ОПК-2	Отчет по индивидуальному заданию, Реферат
	Проработка лекционного материала	4		
	Итого	10		
Итого за семестр		114		
	Подготовка к экзамену	36		Экзамен
Итого		150		

Согласована на портале № 4406

10. Курсовая работа

Не предусмотрено РУП

11. Рейтинговая система для оценки успеваемости студентов

11.1. Балльные оценки для элементов контроля

Таблица 11.1 – Бальные оценки для элементов контроля

Элементы учебной деятельности	Максимальный балл на 1-ую КТ с начала семестра	Максимальный балл за период между 1КТ и 2КТ	Максимальный балл за период между 2КТ и на конец семестра	Всего за семестр
6 семестр				
Коллоквиум		8		8
Контрольная работа	8			8
Опрос на занятиях	8	4	4	16
Отчет по индивидуальному заданию	8	4	4	16
Отчет по лабораторной работе	8	6	8	22
Итого максимум за период	32	22	16	70
Экзамен				30
Нарастающим итогом	32	54	70	100

11.2. Пересчет баллов в оценки за контрольные точки

Пересчет баллов в оценки за контрольные точки представлен в таблице 11.2.

Таблица 11. 2 – Пересчет баллов в оценки за контрольные точки

Баллы на дату контрольной точки	Оценка
≥ 90% от максимальной суммы баллов на дату КТ	5
От 70% до 89% от максимальной суммы баллов на дату КТ	4
От 60% до 69% от максимальной суммы баллов на дату КТ	3
< 60% от максимальной суммы баллов на дату КТ	2

11.3. Пересчет суммы баллов в традиционную и международную оценку

Пересчет суммы баллов в традиционную и международную оценку представлен в таблице 11.3.

Таблица 11. 3 – Пересчет суммы баллов в традиционную и международную оценку

Оценка (ГОС)	Итоговая сумма баллов, учитывает успешно сданный экзамен	Оценка (ECTS)
5 (отлично) (зачтено)	90 - 100	A (отлично)
4 (хорошо) (зачтено)	85 - 89	B (очень хорошо)
	75 - 84	C (хорошо)
	70 - 74	D (удовлетворительно)
3 (удовлетворительно) (зачтено)	65 - 69	
	60 - 64	E (посредственно)

Согласована на портале № 4406

2 (неудовлетворительно) (не зачтено)	Ниже 60 баллов	F (неудовлетворительно)

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Орлов, Сергей Александрович. Технологии разработки программного обеспечения: современный курс по программной инженерии [Текст] : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - СПб. : ПИТЕР, 2012. - 608 с : ил. - (Учебник для вузов) (Стандарт третьего поколения). - Библиогр.: с. 596-600. - Алф. указ.: с. 601-608. - ISBN 978-5-459-01101-2 (наличие в библиотеке ТУСУР - 15 экз.)

12.2. Дополнительная литература

1. Технология разработки программного обеспечения: Учебное пособие / Калайда В. Т., Романенко В. В. - 2012. 220 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2076, свободный.

12.3. Учебно-методическое пособие и программное обеспечение

1. Технологии программирования: Руководство к организации самостоятельной работы / Кручинин В. В. - 2012. 53 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2845, свободный.
2. Технологии программирования: Учебное пособие / Кручинин В. В. - 2013. 271 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2834, свободный.
3. Языки и технологии программирования: Методические рекомендации к практическим занятиям, лабораторным работам и самостоятельной работе / Надреев И. И. - 2012. 16 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2540, свободный.
4. Технология разработки программных систем: Учебное пособие / Боровской И. Г. - 2012. 260 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2436, свободный.

12.4. Базы данных, информационно справочные и поисковые системы

1. http://agilerussia.ru/
2. http://infostart.ru/public/318229/
3. https://habrahabr.ru/hub/agile/
4. https://habrahabr.ru/company/edison/blog/269789/

13. Материально-техническое обеспечение дисциплины

Требования к аудиториям (помещениям, местам) для проведения занятий:

- компьютерный класс (20 компьютеров);
- проектор (интерактивная доска) для демонстрации презентаций.

Требования к оборудованию рабочих мест преподавателя и обучающихся:

Видеопроектор, ноутбук, переносной экран. В компьютерном классе должны быть установлены операционные системы, Microsoft® Visual Studio® 2005, средства MS Office; Word,PowerPoint.

14. Фонд оценочных средств

Фонд оценочных средств приведен в приложении 1.

15. Методические рекомендации по организации изучения дисциплины

Без рекомендаций.

Согласована на портале № 4406

Приложение 1

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____ П. Е. Троян
«___» _____ 20__ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Технология программирования

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки (специальность): **09.03.01 Информатика и вычислительная техника**
Направленность (профиль): **Автоматизированное управление бизнес-процессами и финансами**
Форма обучения: **очная**
Факультет: **ФВС, Факультет вычислительных систем**
Кафедра: **ЭМИС, Кафедра экономической математики, информатики и статистики**
Курс: **3**
Семестр: **6**

Учебный план набора 2015 года

Разработчики:

- доцент каф. ЭМИС Зариковская Н. В.

Экзамен: 6 семестр

Томск 2016

Согласована на портале № 4406

1. Введение

Фонд оценочных средств (ФОС) является приложением к рабочей программе дисциплины (практики) и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения студентом установленных результатов обучения.

ФОС по дисциплине (практике) используется при проведении текущего контроля успеваемости и промежуточной аттестации студентов.

Перечень закрепленных за дисциплиной (практикой) компетенций приведен в таблице 1.

Таблица 1 – Перечень закрепленных за дисциплиной компетенций

Код	Формулировка компетенции	Этапы формирования компетенций
ОПК-1	Способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем.	Должен знать основные принципы, базовые концепции технологий программирования, основные этапы и принципы создания программного продукта, проблемы сложности, основы проектирования с учетом изменений, обработка исключений, технологию разработки алгоритмов и программ; методы отладки и решения задач на ЭВМ в различных режимах; различия между спецификацией и реализацией, основные стандарты в области инфокоммуникационных систем и технологий, в том числе стандарты единой системы программной документации (ЕСПД); ; Должен уметь ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы; использовать прикладные системы программирования; разрабатывать основные программные документы; ; Должен владеть методами навыками владения одной из технологий программирования; навыками разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня.;
ОПК-2	Способность осваивать методики использования программных средств для решения практических задач.	

Общие характеристики показателей и критериев оценивания компетенций на всех этапах приведены в таблице 2.

Таблица 2 – Общие характеристики показателей и критериев оценивания компетенций по этапам

Показатели и критерии	Знать	Уметь	Владеть
Отлично (высокий уровень)	Обладает фактическими и теоретическими знаниями в пределах изучаемой области с пониманием границ применимости	Обладает диапазоном практических умений, требуемых для развития творческих решений, абстрагирования проблем	Контролирует работу, проводит оценку, совершенствует действия работы

Согласована на портале № 4406

Хорошо (базовый уровень)	Знает факты, принципы, процессы, общие понятия в пределах изучаемой области	Обладает диапазоном практических умений, требуемых для решения определенных проблем в области исследования	Берет ответственность за завершение задач в исследовании, приспосабливает свое поведение к обстоятельствам в решении проблем
Удовлетворительно (пороговый уровень)	Обладает базовыми общими знаниями	Обладает основными умениями, требуемыми для выполнения простых задач	Работает при прямом наблюдении

2 Реализация компетенций

2.1 Компетенция ОПК-1

ОПК-1: Способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем..

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания представлены в таблице 3.

Таблица 3 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	принципы организации, установки, конфигурирования системных программных средств, сетевого ПО	выбирать и квалифицированно применять современные аппаратные и программные средства для реализации информационных и автоматизированных систем	системными инструментальными и сервисными средствами, встроенными средствами управления в ОС Microsoft Windows и Unix
Виды занятий	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Лабораторные занятия; • Самостоятельная работа;
Используемые средства оценивания	• Контрольная работа; • Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Опрос на занятиях; • Коллоквиум; • Реферат; • Экзамен;	• Контрольная работа; • Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Опрос на занятиях; • Коллоквиум; • Реферат; • Экзамен;	• Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Коллоквиум; • Реферат; • Экзамен;

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 4.

Согласована на портале № 4406

Таблица 4 – Показатели и критерии оценивания компетенции на этапах

Состав	Знать	Уметь	Владеть
Отлично (высокий уровень)	• принципы организации, установки, конфигурирования системных программных средств, сетевого ПО ; • принципы и методы организации системных программных средств. Информационные технологии в системах управления;	• выбирать и квалифицированно применять современные аппаратные и программные средства для реализации информационных и автоматизированных систем ; • использовать технологии виртуализации, операционные системы, программное обеспечение вычислительных сетей;	• системными инструментальными и сервисными средствами, встроенными средствами управления в ОС Microsoft Windows и Unix ; • профессиональными умениями инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
Хорошо (базовый уровень)	• не в полной мере принципы организации, установки, конфигурирования системных программных средств, сетевого ПО ; • не в полной мере принципы и методы организации системных программных средств. Информационные технологии в системах управления;	• не в полной мере выбирать и квалифицированно применять современные аппаратные и программные средства для реализации информационных и автоматизированных систем ; • не в полной мере использовать технологии виртуализации, операционные системы, программное обеспечение вычислительных сетей;	• не в полной мере системными инструментальными и сервисными средствами, встроенными средствами управления в ОС Microsoft Windows и Unix ; • не в полной мере профессиональными умениями инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем;
Удовлетворительно (пороговый уровень)	• иметь представление об основных принципах и методах организации системных программных средств. Информационные технологии в системах управления;	• иметь начальные навыки по использованию технологии виртуализации, операционные системы, программное обеспечение вычислительных сетей;	• общим представлением об инсталлировании программного и аппаратного обеспечения для информационных и автоматизированных систем;

2.2 Компетенция ОПК-2

ОПК-2: Способность осваивать методики использования программных средств для решения практических задач..

Для формирования компетенции необходимо осуществить ряд этапов. Этапы формирования компетенции, применяемые для этого виды занятий и используемые средства оценивания

Согласована на портале № 4406

представлены в таблице 5.
Таблица 5 – Этапы формирования компетенции и используемые средства оценивания

Состав	Знать	Уметь	Владеть
Содержание этапов	современные объектно-ориентированные алгоритмические языки, их области применения и особенности основные приемы сборочного программирования методики оценки качества объектно-ориентированного проектирования программных систем основные проектные процедуры и технологии при объектно-ориентированом анализе и реализации программных систем ознакомление с дополнительной литературой и передовыми научно-техническими достижениями в области программирования	использовать объектно-ориентированные методы и средства разработки алгоритмов и программ, способы отладки, испытания и документирования программ использовать современные готовые библиотеки классов использовать основные модели, методы и средства информационных технологий и способы их применения для решения задач в предметных областях использовать современные системные программные средства, технологии и инструментальные средства	навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ навыками объектно-ориентированной разработки программного обеспечения навыками адаптации существующих проектных решений к разрабатываемой или модернизируемой программной системе навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ навыками использования инструментальных средств оценки качества объектных моделей программных систем
Виды занятий	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Практические занятия; • Лабораторные занятия; • Лекции; • Самостоятельная работа; • Подготовка к экзамену;	• Лабораторные занятия; • Самостоятельная работа;
Используемые средства оценивания	• Контрольная работа; • Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Опрос на занятиях; • Коллоквиум; • Реферат; • Экзамен;	• Контрольная работа; • Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Опрос на занятиях; • Коллоквиум; • Реферат; • Экзамен;	• Отчет по лабораторной работе; • Отчет по индивидуальному заданию; • Коллоквиум; • Реферат; • Экзамен;

Согласована на портале № 4406

Формулировка показателей и критериев оценивания данной компетенции приведена в таблице 6.

Таблица 6 – Показатели и критерии оценивания компетенции на этапах

Состав	Знать	Уметь	Владеть
Отлично (высокий уровень)	• современные объектно-ориентированные алгоритмические языки, их области применения и особенности; • основные приемы сборочного программирования; • методики оценки качества объектно-ориентированного проектирования программных систем; • основные проектные процедуры и технологии при объектно-ориентированом анализе и реализации программных систем; • ознакомление с дополнительной литературой и передовыми научно-техническими достижениями в области программирования;	• использовать объектно-ориентированные методы и средства разработки алгоритмов и программ, способы отладки, испытания и документирования программ; • использовать современные готовые библиотеки классов; • использовать основные модели, методы и средства информационных технологий и способы их применения для решения задач в предметных областях; • использовать современные системные программные средства, технологии и инструментальные средства;	• навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ; • навыками объектно-ориентированной разработки программного обеспечения; • навыками адаптации существующих проектных решений к разрабатываемой или модернизируемой программной системе; • навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ; • навыками использования инструментальных средств оценки качества объектных моделей программных систем;
Хорошо (базовый уровень)	• методики оценки качества объектно-ориентированного проектирования программных систем; • основные проектные процедуры и технологии при объектно-ориентированом анализе и реализации программных систем; • наличие не искажающих существа ответа погрешностей и пробелов при изложении материала;	• осуществлять объектно-ориентированный анализ предметной области и объектное проектирование структуры программной системы; • использовать объектно-ориентированные методы и средства разработки алгоритмов и программ, способы отладки, испытания и документирования программ;	• навыками восприятия документации и исходного кода объектно-ориентированных программных систем; • навыками объектно-ориентированной разработки программного обеспечения; • навыками адаптации существующих проектных решений к разрабатываемой или модернизируемой программной системе;

Согласована на портале № 4406

		• выделять и повторно использовать проектные решения и программный код в объектной парадигме; • использовать современные готовые библиотеки классов;	• навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ;
Удовлетворительно (пороговый уровень)	• основные проектные процедуры и технологии при объектно-ориентированом анализе и реализации программных систем; • основные теоретические и практические вопросы программного материала;	• использовать объектно-ориентированные методы и средства разработки алгоритмов и программ, способы отладки, испытания и документирования программ;	• навыками применения объектно-ориентированных подходов и инструментальных средств разработки алгоритмов и программ; • навыками объектно-ориентированной разработки программного обеспечения;

3 Типовые контрольные задания

Для реализации вышеперечисленных задач обучения используются типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы, в следующем составе.

3.1 Темы рефератов

– Основные виды тестирования. Тестирование "белого ящика" на стадии кодирования. Регрессионное тестирование. Тестирование "черного ящика".

– Документация по сопровождению программных средств. Стандарт ISO 9126. Модель качества. Характеристики и субхарактеристики качества программного средства. Метрики качества программного средства.

3.2 Темы коллоквиумов

– 1. Что такое информационная среда программы?
– 2. Что такое программное средство (ПС)?
– 3. Что такое ошибка в ПС?
– 4. Что такое надежность ПС?
– 5. Что такое технология программирования?
– 6. Что такое простая и сложная системы?
– 7. Что такое малая и большая системы?
– 8. Что такое жизненный цикл программного средства (ПС)?
– 9. Что такое внешнее описание ПС?
– 10. Что такое сопровождение ПС?
– 11. Что такое качество ПС?
– 12. Что такое смежный контроль?
– 13. Что такое определение требований к программному средству (ПС)?
– 14. Что такое спецификации качества ПС?
– 15. Что такое устойчивость (robustness) ПС?
– 16. Что такое защищенность (defensiveness) ПС?
– 17. Что такое коммуникабельность (communicativeness) ПС?
– 18. Что такое функциональная спецификация ПС?

Согласована на портале № 4406

– 19. Что такое ручная имитация внешнего описания ПС?
– 20. Что такое архитектура программного средства?
– 21. Что такое архитектурная функция?
– 22. Что такое программный модуль?
– 23. Что такое прочность программного модуля?
– 24. Что такое сцепление программного модуля?
– 25. Что такое структурное программирование?
– 6. Что такое пошаговая детализация программного модуля?
– 27. Что такое псевдокод?
– 28. Что такое триада Хоора?
– 29. Что такое свойство программы?
– 30. Пусть заданы описания const n= <конкретное целое значение>; var k, m: integer; x: array[1..n] of integer; Доказать свойство программы: {n>0} m:= x[1] k:=1; ПОКА k<n ДЕЛАТЬ k:= k+1; ЕСЛИ x[k]<m ТО m:= x[k] ВСЕ ЕСЛИ ВСЕ ПОКА; {n>0 & m<= x[i] для всех i, 1<=i<= n}
– 31. Что такое отладка программного средства?
– 32. Что такое тестирование программного средства?
– 33. Что такое автономная отладка программного средства?
– 34. Что такое комплексная отладка программного средства?
– 35. Что такое ведущий отладочный модуль?
– 36. Что такое отладочный имитатор программного модуля?
– 37. Что такое защитное программирование?
– 38. Какие виды защиты программного средства от искажения информации Вы знаете?
– 39. Какие требования предъявляются к компьютеру, чтобы можно было обеспечить защиту программы от отказов другой программы в мультипрограммном режиме?
– 40. Что такое компьютерная подпись?
– 41. Что такое компьютерная печать?
– 42. Какие задачи приходиться решать при обеспечении коммуникабельности ПС?
– 43. Какие возможности предоставляет пользователю графический пользовательский интерфейс?
– 44. Как нужно действовать для обеспечения эффективности ПС?
– 45. Что такое инсталятор программного средства (ПС)?
– 46. Что такое управление конфигурацией ПС?
– 47. Что такое ядро ПС?
– 48. Что такое оболочка ПС?
– 49. Что такое менеджер программного средства?
– 50. Что такое ординарный пользователь программного средства?
– 51. Что такое администратор программного средства?
– 52. Что такое руководство по инсталляции программного средства?
– 53. Что такое руководство по управлению программным средством?
– 54. Что такое управление разработкой ПС?
– 55. Что такое менеджер программного проекта?
– 56. Что такое неформальная демократическая бригада разработчиков ПС?
– 57. Что такое бригада ведущего программиста?
– 58. Что такое смотр программной компоненты (программного документа)?
– 59. Что такое аттестация ПС?

3.3 Темы индивидуальных заданий

– Имеется база данных под управлением MS SQL 2008. В ней находится единственная таблица с полями (Id, Value). В таблице единственное значение (1, "Hello World Revenge"). Необходимо реализовать WCF-сервис, который получит доступ к этой БД через LINQ2SQL, вытянет эту строчку, отрендерит из неё картинку (обычный PNG, в котором на белом фоне будет

Согласована на портале № 4406

написана строчка произвольно выбранным шрифтом), после чего вернет ее как поток. Создать WPF-графическое приложение, в котором будет ссылка этот сервис, которое по нажатию кнопки будет обращаться к сервису, получать картинку и отображать ее на экране.

– Некоторой организации понадобилась система работы с документами: заявлениями, квитанциями и прочими. Основной use-case: пользователь может зайти на страничку, увидеть список документов, которые уже есть в системе, скачать каждый документ в виде PDF, картинки или HTML. Второй основной use-case: пользователь может выбрать какой документ ему создать (из списка всех доступных документов), ввести данные необходимых полей и сохранить документ в системе, после чего он появится в списке и его можно будет скачать, как и все остальные. Узкий момент в том, что заказчик еще сам не знает, какие документы будут в системе, так что архитектурно надо предусмотреть возможность быстро добавить поддержку еще одного документа. Программисты решили сделать сначала прототип без авторизации пользователей и редактирования документов, а так же без дизайна. В качестве наиболее подходящего технологического стека были выбраны ASP.NET MVC, MongoDB и Stimulsoft Reports. Для фронтенда планируется использовать Twitter Bootstrap и jQuery. Подумав, программисты решили отдать прототип данного проекта на реализацию студенту ФИТ, т.к. задача довольно простая.

– Некому брокеру очень хочется разработать dashboard для своих клиентов, где они смогут увидеть свои активы, котировки, посмотреть телефон консультанта, адрес отделения на карте, историю сделок за определенные периоды и прочие статистические данные. Брокер заботливо подготовил мокапы web-интерфейса, однако не спешит говорить, откуда разработчикам брать данные для отображения. Говорил что-то про web-сервисы, но обсуждение и согласование подвисло на стадии определения источников данных на стороне заказчика. Несмотря на это печальное обстоятельство, клиенту уже хочется видеть схематичный интерфейс (Twitter Bootstrap) с тестовыми данными. Менеджмент проекта принимает очевидное решение сделать слой доступа к данным на заглушках, возвращающих тестовые данные, в надежде позже просто подменить реализации. Делать это они решили посредством IoC-контейнера (Unity Application Block, при том обязательно на xml-конфигурации). Реализуйте пожелания менеджмента.

3.4 Темы опросов на занятиях

– Введение в технологию программирования, программную инженерию. Понятие программного средства. Технология программирования и основные этапы ее развития. Проблемы разработки сложных программных систем. Жизненный цикл программного средства. Архитектура ПО. Системный анализ при создании ПС.

– Системный подход при разработке ПС. Модели разработки: каскадная, с промежуточным контролем, спиральная и т.д.; CASE и RAD-технологии. Тестирование и оценка качества. Управление проектом, планирование и распределение ресурсов, контроль исполнения сроков

– Использование декомпозиции и абстракции при проектировании ПО. Спецификация процедур и данных. Внешняя и внутренняя спецификации. Декомпозиция задачи. Методы проектирования структуры ПО. Методы защиты программ и данных. Жизненный цикл программного средства.

– Парадигмы программирования: визуальная, функциональная, процедурная, объектно-ориентированная и т.д. Объектно-ориентированная парадигма: понятия объекта, класса объектов; основные понятия объектно-ориентированного программировании (инкапсуляция, наследование и полиморфизм); классы и объекты; интерфейсы и реализация.

– Библиотеки стандартных компонентов, библиотеки объектов. Проектирование интерфейса с пользователем. Структуры диалога; поддержка пользователя; многооконные интерфейсы; примеры реализации интерфейсов с пользователем с использованием графических пакетов. «Заглушки». «Маленькие хитрости» в программировании. Статические, полустатические и динамические типы данных. Простые и составные типы данных, операция квалификации. Технологии распределенных вычислений: RPC, RMI, Corba, DCOM.

– Обзор и классификация средств поддержки коллективной разработки ПО. Программные средства планирования и управления процессом разработки. Сетевые графики и диаграммы рабочего процесса. Сценарии выполнения работ. Применение систем управления документами. CASE-технологии.

Согласована на портале № 4406

– Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. Отладчики. CASE-технология. UML-диаграммы.

– Категории программных ошибок. Типы тестов. Тестирование на этапе планирования. Тестирование на этапе проектирования. Тестирование "белого ящика" на стадии кодирования. Регрессионное тестирование. Тестирование "черного ящика". Разработка тестов.

3.5 Экзаменационные вопросы

– 1. Жизненный цикл программы

– 2. Триада Хоора.

– 3. Создать приложение, которое проверяет пользователя. Программа выдает два случайных числа, по модулю не превосходящих 100, пользователь пишет в специальную форму произведение этих чисел. Программа проверяет, правильно ли пользователь ввел произведение, в зависимости от правильности выдает свой вердикт. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 4. Постановка задачи. Оценка осуществимости

– 5. Структурное программирование. Пошаговая детализация программного модуля.

– 6. Создать программу, которая соединяет все введенные точки линиями попарно в порядке их ввода. Программа должна обладать интерфейсом с возможностью ввода хотя бы 10-ти точек с целыми координатами. Эти точки рисуются, а потом по нажатию кнопки соединяются линиями. Реализовать выбор цвета линий (хотя бы 5 цветов). Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 7. Планирование

– 8. Программный модуль. Прочность программного модуля. Сцепление программного модуля.

– 9. Создать приложение, которое проверяет пользователя. Программа выдает случайную букву английского/русского алфавита, пользователь пишет в специальную форму гласная или же согласная эта буква. Ввод типа буквы можно осуществить, либо с помощью Textbox, либо кнопками (Button). Программа проверяет, правильно ли пользователь ввел тип буквы, в зависимости от правильности выдает свой вердикт. Должна быть реализована защита от ввода пользователем в Textbox чего-то, что не подходит под формат (если вы выбрали вариант с Textbox). Программа должна быть реализована с помощью Windows forms.

– 10. Управление

– 11. Архитектура программного средства. Архитектурная функция

– 12. Создать программу, которая работает с прямоугольниками. В программе должно быть реализовано: класс прямоугольника с конструктором от четырех целых чисел (координаты левого верхнего угла и правого нижнего угла прямоугольника), метод который может узнать накладываются ли два прямоугольника какой-то ненулевой площадью. Программа должна обладать интерфейсом с возможностью ввода хотя бы двух прямоугольников и проверки их наложения. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 13. Тестирование, обеспечение качества

– 14. Функциональная спецификация ПС. Ручная имитация внешнего описания ПС.

– 15. Создать программу, которая может рисовать три типа геометрических объекта: круг, прямоугольник и треугольник. Для круга задается координата центра и радиус, для прямоугольника координаты левого верхнего и нижнего правого угла, для треугольника координаты его трех вершин. Программа должна обладать интерфейсом с возможностью выбора вида фигуры и ее данных. Реализовать выбор цвета заливки фигуры (хотя бы 5 цветов). Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 16. Групповая разработка, управление версиями

– 17. Защищенность (defensiveness) ПС. Коммуникабельность (communicativeness) ПС.

Согласована на портале № 4406

– 18. Дана матрица n*n. Построить одномерный массив из минимальных элементов каждой строки матрицы. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 19. Психология программирования. Организация коллектива разработчиков

– 20. Спецификации качества ПС. Устойчивость (robustness) ПС.

– 21. Дана матрица n*n. вывести ее строки в порядке возрастания их максимальных элементов. если у двух строк одинаковый максимальный элемент сортировать по следующему элементу. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 22. Документирование

– 23. Смежный контроль. Определение требований к программному средству (ПС).

– 24. Реализовать последовательность Фибоначчи. Найти все простые числа полученной последовательности. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 25. Сопровождение

– 26. Сопровождение ПС. Качество ПС.

– 27. Найти простые числа последовательности 1, 2, 3, ... , n (простой поиск или Решето Эратосфена).

– 28. Реинжиниринг

– 29. Жизненный цикл программного средства (ПС). Внешнее описание ПС.

– 30. Реализовать решение простых геометрических задач с треугольниками. (нахождение площади, периметра, равенство, подобие). Треугольники заданы координатами вершин. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 31. Управление качеством

– 32. Простая и сложная системы. Малая и большая системы.

– 33. Дана строка, найти все полиндромы.

– 34. Стандарты ISO

– 35. Ошибки в ПС. Надежность ПС.

– 36. Дана строка, вывести количество вхождений каждой буквы.

– 37. CASE-технологии. Развитие методологии проектирования

– 38. Информационная среда программы. Программное средство (ПС).

– 39. Дан набор символов, можно ли составить из них данные слова. (входной файл: первая строка - строка-набор символов, вторая строка - число n - число слов, которые необходимо проверить, следующие строки - слова).

– 40. Свойство программы.

– 41. Реализовать калькулятор перевода числа из системы счисления A в систему счислиния B. (A и B до 35).

– 42. Защитное программирование. Виды защиты программного средства от искажения информации. Требования предъявляются к компьютеру, чтобы можно было обеспечить защиту программы от отказов другой программы в мультипрограммном режиме.

– 43. Реализовать сортировку "вставками".

– 44. Пусть заданы описания const n= <конкретное целое значение>; var k, m: integer; x: array[1..n] of integer; Доказать свойство программы: {n>0} m:= x[1] k:=1; ПОКА k<n ДЕЛАТЬ k:= k+1; ЕСЛИ x[k]<m ТО m:= x[k] ВСЕ ЕСЛИ ВСЕ ПОКА; {n>0 & m<= x[i] для всех i, 1<=i<= n}

– 45. Вычислить площадь фигуры, заданной точками.

– 46. Автономная отладка программного средства. Комплексная отладка программного средства.

– 47. Реализовать сортировку "пузырьком".

– 48. Ведущий отладочный модуль. Отладочный имитатор программного модуля.

– 49. Реализовать сортировку "выбором".

– 50. Отладка программного средства. Тестирование программного средства.

Согласована на портале № 4406

– 51. Реализовать матричный калькулятор (на вход дается 2 матрицы и действие, которое необходимо произвести).

– 52. Компьютерная подпись. Компьютерная печать.

– 53. Задачи, решаемые при обеспечении коммуникабельности ПС. Возможности, предоставляемые пользователю графический пользовательский интерфейс.

– 54. Эффективность ПС. Инсталлятор программного средства (ПС).

– 55. Управление конфигурацией ПС. Ядро ПС. Оболочка ПС.

– 56. Создать программу, которая работает с прямоугольниками. В программе должно быть реализовано: класс прямоугольника с конструктором от четырех целых чисел (координаты левого верхнего угла и правого нижнего угла прямоугольника), метод, который может узнать накладываются ли два прямоугольника какой-то ненулевой площадью. Программа должна обладать интерфейсом с возможностью ввода хотя бы двух прямоугольников и проверки их наложения. Должна быть реализована защита от ввода пользователем не тех символов, т.е. не цифр. Программа должна быть реализована с помощью Windows forms.

– 57. Менеджер программного средства. Ординарный пользователь программного средства. Администратор программного средства.

– 58. Управление разработкой ПС. Менеджер программного проекта.

3.6 Темы контрольных работ

– Построить сетевой график написания курсовой работы по теме создание базы данных (указать тему).

– Тестирование, обеспечение качества

– Управлением качеством ПО

– Планирование

3.7 Темы лабораторных работ

– Инструментальная среда разработки

– Классы. Создание класса. Сбор мусора

– Средства отладки и обработка ошибок

– Добавление компонентов и реализация членов

– Применение производных классов

– Подключение к базе данных

– Создание нестандартных элементов управления

– Создание пользовательского интерфейса, меню

– Технологические средства разработки

4 Методические материалы

Для обеспечения процесса обучения и решения задач обучения используются следующие материалы:

– методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы фор-мирования компетенций, согласно п. 12 рабочей программы.

4.1. Основная литература

1. Орлов, Сергей Александрович. Технологии разработки программного обеспечения: современный курс по программной инженерии [Текст] : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - СПб. : ПИТЕР, 2012. - 608 с : ил. - (Учебник для вузов) (Стандарт третьего поколения). - Библиогр.: с. 596-600. - Алф. указ.: с. 601-608. - ISBN 978-5-459-01101-2 (наличие в библиотеке ТУСУР - 15 экз.)

4.2. Дополнительная литература

1. Технология разработки программного обеспечения: Учебное пособие / Калайда В. Т., Романенко В. В. - 2012. 220 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2076, свободный.

Согласована на портале № 4406

4.3. Учебно-методическое пособие и программное обеспечение

1. Технологии программирования: Руководство к организации самостоятельной работы / Кручинин В. В. - 2012. 53 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2845, свободный.
2. Технологии программирования: Учебное пособие / Кручинин В. В. - 2013. 271 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2834, свободный.
3. Языки и технологии программирования: Методические рекомендации к практическим занятиям, лабораторным работам и самостоятельной работе / Надреев И. И. - 2012. 16 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2540, свободный.
4. Технология разработки программных систем: Учебное пособие / Боровской И. Г. - 2012. 260 с. [Электронный ресурс] - Режим доступа: http://edu.tusur.ru/publications/2436, свободный.

4.4. Базы данных, информационно справочные и поисковые системы

1. http://agilerussia.ru/
2. http://infostart.ru/public/318229/
3. https://habrahabr.ru/hub/agile/
4. https://habrahabr.ru/company/edison/blog/269789/

Согласована на портале № 4406