МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

УТВЕРЖДАЮ
Директор департамента образования
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Технология программирования

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **09.03.01 Информатика и вычислительная техника**
Направленность (профиль) / специализация: **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 работы	34	34	часов
4	Всего аудиторных занятий	102	102	часов
5	Самостоятельная работа	78	78	часов
6	Всего (без экзамена)	180	180	часов
7	Подготовка и сдача экзамена	36	36	часов
8	Общая трудоемкость	216	216	часов
		6.0	6.0	З.Е.

Экзамен: 6 семестр

Томск 2018

Согласована на портале № 38657

Рассмотрена и одобрена на заседании кафедры
протокол № 11 от «21» 5 2018 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника, утвержденного 12.01.2016 года, рассмотрена и одобрена на заседании кафедры ЭМИС «___» _____ 20__ года, протокол №_____.

Разработчик:

доцент каф. ЭМИС	_____	Н. В. Зариковская
Заведующий обеспечивающей каф. ЭМИС	_____	И. Г. Боровской

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФВС	_____	Л. А. Козлова
Заведующий выпускающей каф. ЭМИС	_____	И. Г. Боровской

Эксперты:

доцент кафедры ЭМИС	_____	Е. А. Шельмина
Профессор кафедры экономической математики, информатики и статистики (ЭМИС)	_____	С. И. Колесникова

Согласована на портале № 38657

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.5) относится к блоку 1 (вариативная часть).

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

Последующими дисциплинами являются: Управление проектами.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

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

– ПК-2 Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.;

В результате изучения дисциплины обучающийся 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** 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 6.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры
		6 семестр

Согласована на портале № 38657

Аудиторные занятия (всего)	102	102
Лекции	34	34
Практические занятия	34	34
Лабораторные работы	34	34
Самостоятельная работа (всего)	78	78
Оформление отчетов по лабораторным работам	44	44
Проработка лекционного материала	20	20
Подготовка к практическим занятиям, семинарам	14	14
Всего (без экзамена)	180	180
Подготовка и сдача экзамена	36	36
Общая трудоемкость, ч	216	216
Зачетные Единицы	6.0	6.0

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	Лек., ч	Прак. зан., ч	Лаб. раб., ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
6 семестр						
1 Вводная часть	2	0	0	1	3	ОПК-2, ПК-2
2 Организация процесса проектирования программного обеспечения	4	4	0	3	11	ОПК-2, ПК-2
3 Методы проектирования программного обеспечения	4	4	0	3	11	ОПК-2, ПК-2
4 Парадигмы программирования	4	0	22	31	57	ОПК-2, ПК-2
5 Технология создания программного кода	4	4	8	15	31	ОПК-2, ПК-2
6 Технологии коллективной разработки программного обеспечения	4	6	0	13	23	ОПК-2, ПК-2
7 Технологические средства разработки программного обеспечения	4	4	4	5	17	ОПК-2, ПК-2
8 Методы отладки и тестирования программ	4	6	0	3	13	ОПК-2, ПК-2
9 Документирование и оценка качества программных продуктов	4	6	0	4	14	ОПК-2, ПК-2
Итого за семестр	34	34	34	78	180	
Итого	34	34	34	78	180	

Согласована на портале № 38657

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-2, ПК-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-2, ПК-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-2, ПК-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 и объекты; интерфейсы и реализация.	4	ОПК-2, ПК-2
	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	4	ОПК-2, ПК-2

Согласована на портале № 38657

	квалификации. Технологии распределенных вычислений: RPC, RMI, Corba, DCOM.		
	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-2, ПК-2
	Итого	4	
7 Технологические средства разработки программного обеспечения	Инструментальная среда разработки. Библиотека VCL. Средства поддержки проекта. Отладчики. CASE-технология. UML-диаграммы.	4	ОПК-2, ПК-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-2, ПК-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 качества программного средства. Метрики качества программного средства. Оценивание характеристик качества программных средств.	4	ОПК-2, ПК-2
	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 представлены 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							+	+	
Последующие дисциплины									
1 Управление проектами	+	+				+			+

Согласована на портале № 38657

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, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий				Формы контроля
	Лек.	Прак. зан.	Лаб. раб.	Сам. раб.	
ОПК-2	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Экзамен, Коллоквиум, Отчет по лабораторной работе, Опрос на занятиях, Тест, Реферат
ПК-2	+	+	+	+	Контрольная работа, Отчет по индивидуальному заданию, Экзамен, Коллоквиум, Отчет по лабораторной работе, Опрос на занятиях, Тест, Реферат

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
6 семестр			
4 Парадигмы программирования	Инструментальная среда разработки	2	ОПК-2, ПК-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-2, ПК-2
	Создание пользовательского интерфейса, меню	4	
	Итого	8	
7 Технологические средства разработки программного	Технологические средства разработки	4	ОПК-2, ПК-2
	Итого	4	

Согласована на портале № 38657

обеспечения			
Итого за семестр		34	

8. Практические занятия (семинары)

Наименование практических занятий (семинаров) приведено в таблице 8.1.

Таблица 8.1 – Наименование практических занятий (семинаров)

Названия разделов	Наименование практических занятий (семинаров)	Трудоемкость, ч	Формируемые компетенции
6 семестр			
2 Организация процесса проектирования программного обеспечения	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-2, ПК-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-2, ПК-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-2, ПК-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-2, ПК-2
	Итого	6	
7 Технологические средства разработки программного обеспечения	Инструментальная среда разработки. Библиотека VCL. Отладчики. CASE-технология. UML-диаграммы.	4	ОПК-2, ПК-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-2, ПК-2

Согласована на портале № 38657

	Итого	6	
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 качества программного средства. Метрики качества программного средства.	6	ОПК-2, ПК-2
	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	1	ОПК-2, ПК-2	Опрос на занятиях, Отчет по лабораторной работе, Тест, Экзамен
	Итого	1		
2 Организация процесса проектирования программного обеспечения	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Тест, Экзамен
	Проработка лекционного материала	1		
	Итого	3		
3 Методы проектирования программного обеспечения	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Опрос на занятиях, Тест, Экзамен
	Проработка лекционного материала	1		
	Итого	3		
4 Парадигмы программирования	Проработка лекционного материала	1	ОПК-2, ПК-2	Опрос на занятиях, Отчет по лабораторной работе, Тест, Экзамен
	Оформление отчетов по лабораторным работам	30		
	Итого	31		
5 Технология создания программного кода	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Отчет по лабораторной работе, Тест, Экзамен
	Проработка лекционного материала	1		

Согласована на портале № 38657

	Оформление отчетов по лабораторным работам	12		
	Итого	15		
6 Технологии коллективной разработки программного обеспечения	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Тест, Экзамен
	Проработка лекционного материала	11		
	Итого	13		
7 Технологические средства разработки программного обеспечения	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Опрос на занятиях, Отчет по индивидуальному заданию, Отчет по лабораторной работе, Тест, Экзамен
	Проработка лекционного материала	1		
	Оформление отчетов по лабораторным работам	2		
	Итого	5		
8 Методы отладки и тестирования программ	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Контрольная работа, Опрос на занятиях, Отчет по индивидуальному заданию, Тест, Экзамен
	Проработка лекционного материала	1		
	Итого	3		
9 Документирование и оценка качества программных продуктов	Подготовка к практическим занятиям, семинарам	2	ОПК-2, ПК-2	Отчет по индивидуальному заданию, Реферат, Тест, Экзамен
	Проработка лекционного материала	2		
	Итого	4		
Итого за семестр		78		
	Подготовка и сдача экзамена	36		Экзамен
Итого		114		

10. Курсовой проект / курсовая работа

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

11.1. Балльные оценки для элементов контроля

Таблица 11.1 – Балльные 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

Согласована на портале № 38657

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

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Новые технологии в программировании [Электронный ресурс]: Учебное пособие / Д. В. Гарайс, А. Е. Горяинов, А. А. Калентьев - 2014. 176 с. - Режим доступа: https://edu.tusur.ru/publications/5796 (дата обращения: 10.07.2018).

2. Технология разработки программного обеспечения [Электронный ресурс]: Учебное пособие / В. Т. Калайда, В. В. Романенко - 2012. 220 с. - Режим доступа: https://edu.tusur.ru/publications/2076 (дата обращения: 10.07.2018).

12.2. Дополнительная литература

1. Технология разработки программных систем [Электронный ресурс]: Учебное пособие / И. Г. Боровской - 2012. 260 с. - Режим доступа: https://edu.tusur.ru/publications/2436 (дата обраще-

Согласована на портале № 38657

ния: 10.07.2018).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Технологии программирования [Электронный ресурс]: Руководство к организации самостоятельной работы / Кручинин В. В. - 2012. 53 с. - Режим доступа: http://edu.tusur.ru/publications/2845 (дата обращения: 10.07.2018).
2. Технологии программирования [Электронный ресурс]: Учебное пособие / Кручинин В. В. - 2013. 271 с. - Режим доступа: http://edu.tusur.ru/publications/2834 (дата обращения: 10.07.2018).
3. Языки и технологии программирования [Электронный ресурс]: Методические рекомендации к практическим занятиям, лабораторным работам и самостоятельной работе / Надреев И. И. - 2012. 16 с. - Режим доступа: http://edu.tusur.ru/publications/2540 (дата обращения: 10.07.2018).
4. Теория и технология программирования [Электронный ресурс]: Методические указания по самостоятельной работе / С. А. Панов - 2015. 11 с. - Режим доступа: https://edu.tusur.ru/publications/5015 (дата обращения: 10.07.2018).
5. Теория и технологии программирования [Электронный ресурс]: Методические указания к лабораторным работам / С. А. Панов - 2015. 28 с. - Режим доступа: https://edu.tusur.ru/publications/5014 (дата обращения: 10.07.2018).
6. Технология программирования [Электронный ресурс]: Учебное пособие / Н. В. Зариковская - 2018. 130 с. - Режим доступа: https://edu.tusur.ru/publications/8244 (дата обращения: 10.07.2018).

12.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

12.4. Профессиональные базы данных и информационные справочные 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 и требуемое программное обеспечение

13.1. Общие требования к материально-техническому и программному обеспечению дисциплины

13.1.1. Материально-техническое и программное обеспечение для лекционных занятий

Для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации используется учебная аудитория с количеством посадочных мест не менее 22-24, оборудованная доской и стандартной учебной мебелью. Имеются демонстрационное оборудование и учебно-наглядные пособия, обеспечивающие тематические иллюстрации по лекционным разделам дисциплины.

Согласована на портале № 38657

13.1.2. Материально-техническое и программное обеспечение для практических занятий

Учебная лаборатория

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, помещение для курсового проектирования (выполнения курсовых работ), помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 424 ауд.

Описание имеющегося оборудования:

- ПЭВМ (Intel Pentium G3440, 3 G, 4 Gb RAM) (12 шт.);
- Магнито-маркерная доска;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-Zip
- Google Chrome
- Microsoft Visual Studio 2012
- Microsoft Windows 7 Pro
- OpenOffice

Лаборатория ГПО / «Лаборатория подготовки разработчиков бизнес-приложений»

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, помещение для курсового проектирования (выполнения курсовых работ), помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 425 ауд.

Описание имеющегося оборудования:

- ПЭВМ (Intel Pentium G3220, 3 G, 4 Gb RAM) (12 шт.);
- Плазменный телевизор;
- Магнито-маркерная доска;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-Zip
- Google Chrome
- Microsoft Visual Studio 2012
- Microsoft Windows 7 Pro
- OpenOffice

13.1.3. Материально-техническое и программное обеспечение для лабораторных работ

Учебная лаборатория

учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, помещение для курсового проектирования (выполнения курсовых работ), помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 424 ауд.

Описание имеющегося оборудования:

- ПЭВМ (Intel Pentium G3440, 3 G, 4 Gb RAM) (12 шт.);
- Магнито-маркерная доска;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Согласована на портале № 38657

Программное обеспечение:
- 7-Zip
- Google Chrome
- Microsoft Visual Studio 2012
- Microsoft Windows 7 Pro
- OpenOffice

13.1.4. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися **с нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися **с нарушениями зрениями** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися **с нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

1) Что означает «Описать процесс»:

Согласована на портале № 38657

1. определить последовательность состояний заданной информационной среды (правильный).

2. расписать данный процесс по пунктам и составляющим.

3. показать последовательность состояний заданной информационной среды.

2) Дейкстра выделил три интеллектуальные возможности человека, используемые при разработке программных средств. Какие?

1. Способность к перебору, способность к математической индукции, способность к интеллектуальному развитию.

2. способность к перебору, способность к абстракции, способность к математической индукции (правильный).

3. Способность к абстракции, способность к математической логике, способность к предугадыванию событий.

3) Понятие «Функциональность»:

1. количество функций в данном программном средстве.

2. способность программного средства выполнять набор функций, удовлетворяющих любые потребности пользователя.

3. способность программного средства выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей (правильный).

4) Методы борьбы со сложностью систем:

1. обеспечения независимости компонент системы, использование в системах иерархических структур (правильный).

2. обеспечения независимости компонент системы.

3. использование в системах иерархических структур (правильный).

5) Процесс перевода (этапы):

1. составьте план, поймите задачу, выполните план, проанализируйте полученное решение.

2. поймите задачу, составьте план, выполните план, проанализируйте полученное решение (правильный).

3. составьте план, выполните план, проанализируйте полученное решение, поймите задачу.

6) Структуру внешнего описания программного средства можно выразить формулой:

1. внешнее описание ПС = определение требований - спецификация качества ПС + функциональная спецификация ПС.

2. внешнее описание ПС = определение требований - спецификация качества ПС - функциональная спецификация ПС.

3. внешнее описание ПС = определение требований + спецификация качества ПС + функциональная спецификация ПС (правильный).

7) Три способа разработки определения требований к программному средству:

1. управляемая пользователем разработка, контролируемая пользователем разработка, независимая от пользователя разработка (правильный).

2. управляемая пользователем разработка, неуправляемая пользователем разработка, независимая от пользователя разработка.

3. контролируемая пользователем разработка, неконтролируемая пользователем разработка, управляемая пользователем разработка

8) Понятие «Расширяемость»:

1. свойство, характеризующее способность ПС к использованию большего объема памяти для хранения данных или расширению функциональных возможностей отдельных компонент (правильный).

2. свойство, характеризующее способность ПС к использованию маленького объема памяти для хранения данных или расширению функциональных возможностей отдельных компонент.

3. свойство, характеризующее способность ПС к использованию маленького объема памяти для хранения данных.

9) Методы контроля внешнего описания программного средства:

1. статический просмотр, смежный контроль.

2. статический просмотр, смежный контроль, пользовательский контроль, ручная имитация (правильный).

Согласована на портале № 38657

3. пользовательский контроль, ручная имитация.

10) Основные задачи разработки архитектуры ПС:

1. Выделение программных подсистем и отображение на них внешних функций ПС, определение способов взаимодействия между выделенными программными подсистемами (правильный).

2. Выделение программных подсистем и отображение на них внешних функций ПС.

3. определение способов взаимодействия между выделенными программными подсистемами.

11) Основные классы архитектур программных средств:

1. цельная программа, комплекс автономно выполняемых программ, слоистая программная система, коллектив параллельно выполняемых программ (правильный).

2. цельная программа, комплекс автономно выполняемых программ.

3. слоистая программная система, коллектив параллельно выполняемых программ.

12) Понятие «Программный модуль»:

1. конкретный фрагмент описания процесса, оформляемый как самостоятельный программный продукт, непригодный для использования в описаниях процесса.

2. конкретный фрагмент описания процесса, оформляемый как самостоятельный программный продукт, пригодный для использования в описаниях процесса.

3. любой фрагмент описания процесса, оформляемый как самостоятельный программный продукт, пригодный для использования в описаниях процесса (правильный).

13) Понятие «Прочность модуля»:

1. мера его внутренних связей (правильный).

2. мера его внешних связей.

3. Мера его внешних и внутренних связей.

14) Понятие «Сцепление модуля»:

1. мера его зависимости по данным отдельно взятых модулей, а также своего модуля.

2. мера его зависимости по данным своего модуля.

3. мера его зависимости по данным от других модулей (правильный).

15) Для контроля структуры программы можно использовать три метода:

1. статический контроль, смежный контроль, сквозной контроль (правильный).

2. смежный контроль, статический контроль, параллельный контроль.

3. сквозной контроль, статический контроль, фоновый контроль.

16) При разработке программного модуля целесообразно придерживаться следующего порядка:

1. выбор языка программирования, изучение и проверка спецификации модуля, выбор алгоритма и структуры данных, программирование (кодирование) модуля, шлифовка текста модуля, проверка модуля, компиляция модуля.

2. изучение и проверка спецификации модуля, выбор языка программирования, выбор алгоритма и структуры данных, программирование (кодирование) модуля, шлифовка текста модуля, проверка модуля, компиляция модуля (правильный).

3. изучение и проверка спецификации модуля, выбор алгоритма и структуры данных, выбор языка программирования, программирование (кодирование) модуля, шлифовка текста модуля, проверка модуля, компиляция модуля.

17) Применяются следующие методы контроля программного модуля:

1. сквозное прослеживание, доказательство свойств программного модуля.

2. статическая проверка текста модуля, сквозное прослеживание, доказательство свойств программного модуля (правильный).

3. статическая проверка текста модуля, сквозное прослеживание.

18) Понятие «Отладка программного средства»:

1. деятельность, направленная на обнаружение и исправление ошибок в программном средстве с использованием процессов выполнения его программ (правильный).

2. деятельность, направленная на обнаружение в программном средстве с использованием процессов выполнения его программ.

3. деятельность, направленная на исправление ошибок в программном средстве с использо-

Согласована на портале № 38657

ванием процессов выполнения его программ.

19) Управление разработкой программного средства:

1. деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков ПС, на планирование и контроль деятельности этого коллектива с целью обеспечения требуемого качества ПС, выполнения сроков и бюджета разработки ПС (правильный).

2. деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков ПС, выполнения сроков и бюджета разработки ПС.

3. деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков ПС, на планирование и контроль деятельности этого коллектива с целью обеспечения требуемого качества ПС

20) Управление издержками по разработке программного средства:

1. деятельность, направленная на обеспечение подходящего времени для разработки в рамках выделенного бюджета.

2. деятельность, направленная на обеспечение подходящей стоимости разработки в рамках выделенного бюджета (правильный).

3. деятельность, направленная на обеспечение подходящего места разработки в рамках выделенного бюджета.

14.1.2. 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.

Согласована на портале № 38657

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.

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}

Согласована на портале № 38657

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.

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.

14.1.3. Темы опросов на занятиях

• 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.

• 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

• 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

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

14.1.4. 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 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.

Согласована на портале № 38657

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.

14.1.5. 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?
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?

Согласована на портале № 38657

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?

14.1.6. 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.

14.1.7. Темы контрольных работ

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

Тестирование, обеспечение качества

Управлением качеством ПО

Планирование

14.1.8. Темы лабораторных работ

Инструментальная среда разработки

Классы. Создание класса. Сбор мусора

Средства отладки и обработка ошибок

Согласована на портале № 38657

Добавление компонентов и реализация членов
Применение производных классов
Подключение к базе данных
Создание нестандартных элементов управления
Создание пользовательского интерфейса, меню
Технологические средства разработки

14.2. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами исходя из состояния обучающегося на момент проверки

14.3. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

- в печатной форме;
- в печатной форме с увеличенным шрифтом;
- в форме электронного документа;
- методом чтения ассистентом задания вслух;
- предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

- письменно на бумаге;
- набор ответов на компьютере;
- набор ответов с использованием услуг ассистента;
- представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в форме электронного документа;
- в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:

- в форме электронного документа;

Согласована на портале № 38657

– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

– в форме электронного документа;

– в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 38657