МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ 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

Системное программное обеспечение

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **15.03.04 Автоматизация технологических процессов и производств**
Направленность (профиль) / специализация: **Системы автоматизации технологических процессов и производств**
Форма обучения: **очная**
Факультет: **ФВС, Факультет вычислительных систем**
Кафедра: **КСУП, Кафедра компьютерных систем в управлении и проектировании**
Курс: **3**
Семестр: **5**
Учебный план набора 2017 года

Распределение рабочего времени

№	Виды учебной деятельности	5 семестр	Всего	Единицы
1	Лекции	22	22	часов
2	Лабораторные работы	32	32	часов
3	Всего аудиторных занятий	54	54	часов
4	Самостоятельная работа	126	126	часов
5	Всего (без экзамена)	180	180	часов
6	Подготовка и сдача экзамена	36	36	часов
7	Общая трудоемкость	216	216	часов
		6.0	6.0	З.Е.

Экзамен: 5 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 23.08.2017
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Томск

Согласована на портале № 48018

Рассмотрена и одобрена на заседании кафедры
протокол № 17 от «30» 5 2019 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 15.03.04 Автоматизация технологических процессов и производств, утвержденного 12.03.2015 года, рассмотрена и одобрена на заседании кафедры КСУП «___» _____ 20__ года, протокол №_____.

Разработчик:

Доцент каф. КСУП _____ А. Е. Горяинов

Заведующий обеспечивающей каф. КСУП _____ Ю. А. Шурыгин

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФВС _____ М. В. Черкашин

Заведующий выпускающей каф. КСУП _____ Ю. А. Шурыгин

Эксперты:

Профессор кафедры компьютерных систем в управлении и проектировании (КСУП) _____ В. М. Зюзьков

Доцент кафедры компьютерных систем в управлении и проектировании (КСУП) _____ Н. Ю. Хабибулина

Согласована на портале № 48018

1. Цели и задачи дисциплины

1.1. Цели дисциплины

Освоение процесса разработки программного обеспечения.

Обучение разработке пользовательских приложений для работы с базами данных.

1.2. Задачи дисциплины

- Изучение языка C# и платформы .NET Framework.
- Изучение среды разработки Visual Studio.
- Изучение принципов создания пользовательского интерфейса.
- Изучение принципов организации тестирования.
- Изучение написания автоматизированных тестов средствами библиотеки NUnit.
- Изучение системы версионного контроля Git и онлайн-сервиса GitHub.
- Создание собственного репозитория проекта и получение навыков работы в нём.
- Получение навыка сборки установщика приложения и поставки его конечному пользователю.
- Получение навыка написания проектной документации к разрабатываемому приложению.
-

2. Место дисциплины в структуре ОПОП

Дисциплина «Системное программное обеспечение» (Б1.В.ОД.1) относится к блоку 1 (вариативная часть).

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

Последующими дисциплинами являются: Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

- ПК-19 способностью участвовать в работах по моделированию продукции, технологических процессов, производств, средств и систем автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством с использованием современных средств автоматизированного проектирования, по разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления процессами;

В результате изучения дисциплины обучающийся должен:

- **знать** основные этапы разработки программного обеспечения; методологии разработки программного обеспечения; перечень документов, относящихся к технической документации по проекту программного обеспечения; виды и способы организации тестирования; принципы создания пользовательских интерфейсов; принципы и подходы командообразования.
- **уметь** прорабатывать архитектуру приложения; читать техническую документацию, в частности, диаграммы классов, диаграммы вариантов использования; составлять пользовательский интерфейс приложения; вести работу в системах версионного контроля; создавать установщики приложений для поставки программ конечному пользователю; оценивать объём выполняемых работ, их стоимость и время разработки.
- **владеть** языком программирования C# и средой разработки .NET Framework; средой разработки Visual Studio; навыком работы с фреймворком WinForms для создания пользовательских интерфейсов; навыком написания автоматизированных тестов; навыком работы в системах версионного контроля; навыком написания технической документации; навыком сборки установщика приложения.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 6.0 зачетных единицы и представлена в таблице 4.1.

Согласована на портале № 48018

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры
		5 семестр
Аудиторные занятия (всего)	54	54
Лекции	22	22
Лабораторные работы	32	32
Самостоятельная работа (всего)	126	126
Оформление отчетов по лабораторным работам	14	14
Проработка лекционного материала	112	112
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	Лаб. раб., ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
5 семестр					
1 Обзор языка C# и платформы .NET Framework	2	0	8	10	ПК-19
2 Проектирование бизнес-логики приложения	4	8	16	28	ПК-19
3 Разработка пользовательского интерфейса	2	8	16	26	ПК-19
4 Тестирование программного обеспечения	2	4	18	24	ПК-19
5 Техническая документация	4	4	18	26	ПК-19
6 Организация процесса разработки программного обеспечения	2	0	20	22	ПК-19
7 Командообразование	2	0	2	4	ПК-19
8 Инструменты и технологии для организации процесса разработки	2	2	14	18	ПК-19
9 Сборка и поставка программного обеспечения	2	6	14	22	ПК-19
Итого за семестр	22	32	126	180	
Итого	22	32	126	180	

Согласована на портале № 48018

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
5 семестр			
1 Обзор языка C# и платформы .NET Framework	Сравнение языка C# с Си++; создание классов; работа с объектами; работа с коллекциями; инкапсуляция; агрегирование; наследование; интерфейсы и полиморфизм; обработка исключительных ситуаций; делегаты и события	2	ПК-19
	Итого	2	
2 Проектирование бизнес-логики приложения	Определение сущностей предметной области, выделение абстракций; создание иерархии объектов предметной области; критерии качества архитектуры приложения; признаки низкого качества программного кода	4	ПК-19
	Итого	4	
3 Разработка пользовательского интерфейса	Фреимворк WinForms; создание форм и элементов пользовательского интерфейса на языке C#; правила и требования верстки пользовательского интерфейса; шаблоны пользовательского поведения при работе с программным обеспечением; прототипирование	2	ПК-19
	Итого	2	
4 Тестирование программного обеспечения	Цели и описание процесса тестирования; классификация тестов; составление плана тестирования; взаимодействие разработчика и тестировщика; написание автоматизированных модульных тестов с использованием библиотеки NUnit	2	ПК-19
	Итого	2	
5 Техническая документация	Назначение технической документации; стандарты технической документации; виды проектной документации по программному обеспечению; составление и утверждение технического задания; UML-диаграмма классов, UML-диаграмма вариантов использования	4	ПК-19
	Итого	4	
6 Организация процесса разработки программного обеспечения	Обобщенный процесс разработки программного обеспечения; каскадная методология разработки; гибкие методологии разработки: Srcum, Xtreme Programming, Kanban	2	ПК-19
	Итого	2	

Согласована на портале № 48018

7 Командообразование	Проблемы командообразования; разделение в команде по должностным инструкциям; разделение в команде по иерархии и уровню ответственности; командные роли и тест Белбина; трудовая дисциплина	2	ПК-19
	Итого	2	
8 Инструменты и технологии для организации процесса разработки	Системы версионного контроля; системы непрерывной интеграции; инструменты рефакторинга и оптимизации программного кода	2	ПК-19
	Итого	2	
9 Сборка и поставка программного обеспечения	Сборка программы в режиме отладки и релиза; подготовка файлов для сборки установщика; написание скриптов для сборки установщиков в среде InnoSetup; понятие обфускации и защиты исходного кода программы; тестирование установщика	2	ПК-19
	Итого	2	
Итого за семестр		22	

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 Базы данных		+							
2 Объектно-ориентированное программирование	+	+			+			+	
Последующие дисциплины									
1 Защита выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты				+	+	+		+	+

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.	Лаб. раб.	Сам. раб.	

Согласована на портале № 48018

ПК-19	+	+	+	Контрольная работа, Защита отчета, Отчет по лабораторной работе, Опрос на занятиях, Тест, Дифференцированный зачет

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
5 семестр			
2 Проектирование бизнес-логики приложения	Проектирование и реализация бизнес-логики приложения	8	ПК-19
	Итого	8	
3 Разработка пользовательского интерфейса	Разработка пользовательского интерфейса приложения на основе WinForms	8	ПК-19
	Итого	8	
4 Тестирование программного обеспечения	Написание автоматизированных модульных тестов	4	ПК-19
	Итого	4	
5 Техническая документация	Составление проекта системы	4	ПК-19
	Итого	4	
8 Инструменты и технологии для организации процесса разработки	Работа с системами версионного контроля	2	ПК-19
	Итого	2	
9 Сборка и поставка программного обеспечения	Рефакторинг программы и сборка установщика	6	ПК-19
	Итого	6	
Итого за семестр		32	

8. Практические занятия (семинары)

Не предусмотрено РУП.

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
5 семестр				
1 Обзор языка C# и платформы .NET Framework	Проработка лекционного материала	8	ПК-19	Контрольная работа, Опрос на занятиях, Тест
	Итого	8		
2 Проектирование	Проработка лекционно-	10	ПК-19	Дифференцирован-

Согласована на портале № 48018

бизнес-логики приложения	го материала			ный зачет, Защита отчета, Контрольная работа, Опрос на занятиях, Отчет по лабораторной работе, Тест
	Оформление отчетов по лабораторным работам	2		
	Оформление отчетов по лабораторным работам	4		
	Итого	16		
3 Разработка пользовательского интерфейса	Проработка лекционного материала	16	ПК-19	Дифференцированный зачет, Опрос на занятиях, Отчет по лабораторной работе, Тест
	Итого	16		
4 Тестирование программного обеспечения	Проработка лекционного материала	16	ПК-19	Дифференцированный зачет, Защита отчета, Отчет по лабораторной работе, Тест
	Оформление отчетов по лабораторным работам	2		
	Итого	18		
5 Техническая документация	Проработка лекционного материала	16	ПК-19	Дифференцированный зачет, Защита отчета, Контрольная работа, Опрос на занятиях, Отчет по лабораторной работе, Тест
	Оформление отчетов по лабораторным работам	2		
	Итого	18		
6 Организация процесса разработки программного обеспечения	Проработка лекционного материала	20	ПК-19	Дифференцированный зачет, Опрос на занятиях, Тест
	Итого	20		
7 Командообразование	Проработка лекционного материала	2	ПК-19	Дифференцированный зачет, Опрос на занятиях, Тест
	Итого	2		
8 Инструменты и технологии для организации процесса разработки	Проработка лекционного материала	12	ПК-19	Дифференцированный зачет, Защита отчета, Отчет по лабораторной работе, Тест
	Оформление отчетов по лабораторным работам	2		
	Итого	14		
9 Сборка и поставка программного обеспечения	Проработка лекционного материала	12	ПК-19	Дифференцированный зачет, Защита отчета, Отчет по лабораторной работе, Тест
	Оформление отчетов по лабораторным работам	2		
	Итого	14		
Итого за семестр		126		
	Подготовка и сдача экзамена	36		Экзамен
Итого		162		

Согласована на портале № 48018

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
5 семестр				
Дифференцированный зачет			10	10
Защита отчета	5	5	10	20
Контрольная работа	5	5		10
Опрос на занятиях	5	5	5	15
Отчет по лабораторной работе	5	5	5	15
Итого максимум за период	20	20	30	70
Экзамен				30
Нарастающим итогом	20	40	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	Ниже 60 баллов	F (неудовлетворительно)

Согласована на портале № 48018

зачтено)		

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Технология разработки программного обеспечения [Электронный ресурс]: Учебное пособие / Калайда В. Т., Романенко В. В. – 2012. 220 с. — Режим доступа: https://edu.tusur.ru/training/publications/2076 (дата обращения: 28.06.2019).
2. Мирютов А.А. Проектирование программных систем: учебное пособие. Федеральное агентство по образованию, ТУСУР, Кафедра ЭС. –Томск: ТУСУР, 2008. -233с.:ил. (наличие в библиотеке ТУСУР - 100 экз.)
3. Новые технологии в программировании [Электронный ресурс]: Учебное пособие / Гарайс Д. В., Горяинов А. Е., Калентьев А. А. – 2014. 176 с. — Режим доступа: https://edu.tusur.ru/training/publications/5796 (дата обращения: 28.06.2019).

12.2. Дополнительная литература

1. Брауде Э.Д. Технология разработки программного обеспечения / Э.Д. Брауде. – СПб. : Питер, 2004. – 654с. (наличие в библиотеке ТУСУР - 22 экз.)

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Новые технологии в программировании [Электронный ресурс]: Методические указания к лабораторным работам / Гарайс Д. В., Горяинов А. Е., Калентьев А. А. – 2015. 79 с. (Методические указания по выполнению лабораторных работ и самостоятельной работы представлены на стр. 5-75) — Режим доступа: https://edu.tusur.ru/training/publications/5795 (дата обращения: 28.06.2019).

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://msdn.com,
2. http://ieeexplore.ieee.org
3. Дополнительно к профессиональным базам данных рекомендуется использовать информационные, справочные и нормативные базы данных https://lib.tusur.ru/ru/resursy/bazy-dannyh

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

Согласована на портале № 48018

демонстрационное оборудование и учебно-наглядные пособия, обеспечивающие тематические иллюстрации по лекционным разделам дисциплины.

13.1.2. Материально-техническое и программное обеспечение для лабораторных работ

Лаборатория информационного обеспечения систем управления

учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий практического типа, учебная аудитория для проведения занятий лабораторного типа, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

634034, Томская область, г. Томск, Вершинина улица, д. 74, 329 ауд.

Описание имеющегося оборудования:

- Компьютер WS1 (11 шт.);
- Доска белая;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- Enterprise Architect
- Microsoft Visual Studio 2013 Professional
- OpenOffice 4
- Windows 10 Enterprise

13.1.3. 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

Согласована на портале № 48018

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) С какой фундаментальной проблемой при разработке ПО сегодня сталкиваются программисты:

1) Небольшие вычислительные мощности современных ЭВМ

2) Ограниченность высокоуровневых языков программирования

3) Отсутствие необходимых парадигм для разработки сложных программных систем

4) Концептуальная сложность программной системы, которая охватывает большое

(2) Этап разработки ПО, ошибка на котором «стоит» наибольшее количество ресурсов, выделенных на проект:

1) Составление ТЗ и анализ задачи

2) Тестирование ПО

3) Составление проекта программной системы

4) Появление задачи

(3) В каком случае не нужно использовать формальный подход к составлению проекта системы:

1) При реализации небольших проектов

2) При вовлечении в разработку большого количества людей

3) При разработке сложной программной системы

4) При работе в большой команде разработки

(4) В какой из этапов разработки ПО входит этап написания модульных тестов:

1) Разработка ТЗ

2) Кодирование

3) Разработка проекта системы

4) Этап тестирования ПО

(5) Какая из областей подходит для использования каскадной методологии

1) Разработка для решения задач бизнеса

2) Разработка для решения задач государственного сектора

3) Разработка для решения задач науки

4) Разработка для решения задач в области высоконагруженных вычислений

(6) Список требований к функциональности разрабатываемой системы, упорядоченный по степени их важности в методологии Scrum называется^

1) Пожелания пользователя

2) Резерв спринта

3) Резерв проекта

4) Спринт

(7) Что согласно методологии Scrum позволяет сделать процесс разработки ПО гибче

1) Ограничения на резерв проекта

2) Оперативное получение обратной связи от владельца проекта

3) Ведение диаграммы выполнения задач

4) Длительность спринта

(8) К особенностям методологии Kanban не относится:

1) Отсутствие пошагового руководства к действию

Согласована на портале № 48018

2) Концепция: «Уменьшение выполняющейся в данный момент работы»

3) Большая гибкость методологии, чем других: XP и Scrum

4) Концепция парного программирования

(10) IDEF это:

1) Семейство совместно используемых методов для решения задач моделирования сложных систем, позволяет отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах.

2) Унифицированный язык моделирования объектно-ориентированных систем

3) Система УГО для описания работы алгоритма

4) Системы УГО для описания работы динамических систем

(11) Какой из типов UML диаграмм входит в структурные диаграммы:

1) Диаграмма классов

2) Диаграмма вариантов использования

3) Диаграмма деятельности

4) Диаграмма последовательности

(12) Какие связи на диаграмме вариантов использования могут быть между действующими лицами:

1) Связь включения

2) Связь расширения

3) Связь обобщения

4) Связь ассоциации

(13) Какие связи на диаграмме прецедентов не могут быть между вариантами использования:

1) Связь включения

2) Связь расширения

3) Связь обобщения

4) Связь ассоциации

(14) Как в классе обозначаются статические поля и методы

1) Подчёркиванием

2) Курсивом

3) Знаком +

4) Атрибутом static

(15) Какой из ниже перечисленных вопросов является необязательным для рассмотрения в техническом задании?

1) Цель и назначение программы

2) Исходная проблема, решением которой должна являться программа

3) Контекст использования

4) Прототип пользовательского интерфейса

(16) Какой шаблон пользовательского поведения описан ниже?

«Хорошее программное обеспечение позволяет людям пробовать неизвестные функции и возвращать систему в исходное состояние, снова пробовать что-то новое и так далее»

1) Безопасное исследование

2) Мгновенное вознаграждение

3) Разумная достаточность

4) Изменения на полпути

(17) Какой шаблон пользовательского поведения описан ниже?

«Данный шаблон подразумевает, что пользователь скорее выберет достаточно хорошее или удовлетворяющее его решение, а не наилучшее, если изучение всех альтернативных вариантов может потребовать траты времени и сил»

1) Безопасное исследование

2) Мгновенное вознаграждение

3) Разумная достаточность

4) Отложенный выбор

(18) Какой шаблон пользовательского поведения описан ниже?

Согласована на портале № 48018

«Психологический феномен, используемый людьми для напоминания себе о запланированных действиях. Например, если вам нужно отправить письмо через несколько часов (а не прямо сейчас), вы можете оставить себе записку и наклеить её на монитор»

1) Разумная достаточность
2) Изменения на полпути
3) Отложенный выбор
4) Проспективная память

(19) Какой тип прототипа описан ниже?

«Выполняется для проверки архитектурных концепций, затрагивает все уровни реализации»

1) Горизонтальный
2) Вертикальный
3) Исследовательский
4) Электронный

(20) Что такое тестирование?

1) Проверка соответствия объекта желаемым критериям
2) Процесс поиска ошибок в программном продукте
3) Процесс поиска и исправления ошибок в программном продукте
4) Написание тестов и тестовых случаев для разрабатываемого продукта

14.1.2. Экзаменационные вопросы

1. Объекты, классы, сообщения. Интерфейс класса и его реализация. Абстрагирование, инкапсуляция, наследование, полиморфизм. Пример класса на языке C#.

2. Типы данных, литералы и переменные. Типы значений, целочисленные типы, типы для представления чисел с плавающей точкой, символы, логический тип, литералы.

Инициализация переменной, неявно типизированные переменные, область действия и время существования переменных, приведение типов.

3. Операторы: арифметические операторы, операторы отношения и логические операторы, оператор присваивания, тернарный оператор ?

4. Управляющие операторы: if, switch, for, while, do-while, foreach, break, continue, return.

5. Массивы и строки: виды массивов, инициализация массивов, неявно типизированные массивы, постоянство строк.

6. Классы, объекты и методы: общая форма определения класса, создание объектов. Конструкторы, оператор new, сборка мусора. Модификаторы доступа, инициализация объектов. Рекурсия, псевдорекурсия. Статические классы.

7. Методы, использование параметров метода, модификаторы параметров ref и out, использование переменного числа аргументов, перегрузка методов, необязательные аргументы, именованные аргументы, метод Main(), статические методы.

8. Перегрузка операторов: перегрузка унарных и бинарных операторов, операторов отношения, true и false, логических операторов, операторов преобразования.

9. Индексаторы и свойства: одномерные и многомерные индексаторы, перегрузка индексаторов; автосвойства, модификаторы доступа в аксессорах.

10. Наследование: организация защищенного доступа, конструкторы и наследование, сокрытие имен, виртуальные методы, абстрактные классы, ключевое слово sealed, класс object, упаковка и распаковка.

Согласована на портале № 48018

11. Интерфейсы: реализация, применение интерфейсных ссылок, наследование интерфейсов, явные реализации. Структуры и перечисления.

12. Обработка исключительных ситуаций: класс System.Exception, ключевые слова try, catch, throw, finally.

13. Делегаты и события: объявление делегатов, групповая адресация, ковариантность и контрвариантность, анонимные методы, события, групповая адресация события, аксессоры событий.

14. Обобщения: объявление, ограниченные типы, получение значения, присваиваемого параметру типа по умолчанию, обобщенный метод, вариантные делегаты.

15. Этапы разработки программного обеспечения. Общая последовательность, результаты каждого этапа, участники каждого этапа. Виды программной документации.

16. Техническое задание. Назначение документа. Содержание документа и его основных пунктов. Постановка цели и задач. Определение целевой аудитории. Критерии качества продукта.

17. UML-диаграммы. Диаграммы классов. Ассоциация, использование, агрегация, наследование, реализация, обобщение. Кардинальность. Обозначение модификаторов доступа, статических членов класса, виртуальных членов класса на диаграмм классов.

18. UML-диаграммы. Диаграммы активностей. Процессы, передача управления, параллельное выполнение, условия, циклы, суперпроцессы.

19. UML-диаграммы. Диаграммы вариантов использования. Актер, действие, связи: включение, наследование, обобщение, точки обобщения. Назначение и применение диаграмм.

20. Прототипирование. Назначение прототипов. Виды прототипов: горизонтальные и вертикальные, одноразовые и эволюционные, бумажные и электронные.

21. Проектирование интерфейсов. Верстка элементов пользовательского интерфейса.

22. Проектирование интерфейсов. Шаблоны пользовательского поведения: безопасное исследование, мгновенное вознаграждение, разумная достаточность, изменения на полпути, отложенный выбор.

23. Проектирование интерфейсов. Шаблоны пользовательского поведения: пошаговое построение, привыкание, пространственная память, проспективная память, организованное повторение, только клавиатура, советы других людей.

24. Критерии качества спроектированного класса: минимальная сложность, простота сопровождения, слабое сопряжение, расширяемость, возможность повторного использования.

25. Критерии качества спроектированного класса: высокий коэффициент объединения по входу, низкий коэффициент разветвления по выходу, портируемость, достаточная функциональность, стратификация.

26. Оптимизация кода и рефакторинг. Нотация RSDN. Документирование кода.

27. Тестирование. Ошибка, эталон, тест, тестовый случай. Типы тестирования: по знанию внутренней системы, по объекту тестирования.

Согласована на портале № 48018

28. Тестирование. Ошибка, эталон, тест, тестовый случай. Типы тестирования: по степени изолированности тестируемых компонентов, по степени автоматизированности, по степени подготовки к тестированию, по ожидаемому результату.

29. Тестирование. Блочное тестирование: основные принципы, назначение. Способы определения тестовых случаев. Библиотека NUnit. Атрибуты Test, TestCase, Combinatorial, ExpectedException, Repeat, Setup. Класс Assert библиотеки NUnit.

30. Системы версионного контроля. Назначение. Ревизия, репозиторий, рабочая копия. Принципы синхронизации. Конфликты. Ветки и метки.

31. Методологии разработки ПО. Водопадная методология. Преимущества и недостатки. Методология Scrum: общие принципы, виды собраний, основные этапы. Преимущества и недостатки.

32. Методологии разработки ПО. Экстремальное программирование. Основные принципы. Преимущества и недостатки. Методология Kanban, основные принципы, преимущества и недостатки.

14.1.3. Темы контрольных работ

Разработайте архитектуру приложения в виде UML-диаграммы классов для следующей программы:

Программа для автоматического формирования списка литературы в научно-технических публикациях. Программа содержит базу данных библиографических источников: книг, статей журналов и сборников, диссертаций, стандартов и пр. Каждый вид библиографического источника характеризуется собственным набором полей (см. прилаг. ГОСТ). У каждого библиографического источника есть метод по возврату строки с описанием источника, оформленного по ГОСТ

14.1.4. Темы опросов на занятиях

Сравнение языка C# с Си++; создание классов; работа с объектами; работа с коллекциями; инкапсуляция; агрегирование; наследование; интерфейсы и полиморфизм; обработка исключительных ситуаций; делегаты и события

Определение сущностей предметной области, выделение абстракций; создание иерархии объектов предметной области; критерии качества архитектуры приложения; признаки низкого качества программного кода

Фреимворк WinForms; создание форм и элементов пользовательского интерфейса на языке C#; правила и требования верстки пользовательского интерфейса; шаблоны пользовательского поведения при работе с программным обеспечением; прототипирование

Назначение технической документации; стандарты технической документации; виды проектной документации по программному обеспечению; составление и утверждение технического задания; UML-диаграмма классов, UML-диаграмма вариантов использования

Обобщенный процесс разработки программного обеспечения; каскадная методология разработки; гибкие методологии разработки: Srcum, Xtreme Programming, Kanban

Проблемы командообразования; разделение в команде по должностным инструкциям; разделение в команде по иерархии и уровню ответственности; командные роли и тест Белбина; трудовая дисциплина

14.1.5. Вопросы дифференцированного зачета

1. Объекты, классы, сообщения. Интерфейс класса и его реализация. Абстрагирование, инкапсуляция, наследование, полиморфизм. Пример класса на языке C#.

2. Типы данных, литералы и переменные. Типы значений, целочисленные типы, типы для представления чисел с плавающей точкой, символы, логический тип, литералы.

Инициализация переменной, неявно типизированные переменные, область действия и вре-

Согласована на портале № 48018

мя существования переменных, приведение типов.

3. Операторы: арифметические операторы, операторы отношения и логические операторы, оператор присваивания, тернарный оператор ?

4. Управляющие операторы: if, switch, for, while, do-while, foreach, break, continue, return.

5. Массивы и строки: виды массивов, инициализация массивов, неявно типизированные массивы, постоянство строк.

6. Классы, объекты и методы: общая форма определения класса, создание объектов. Конструкторы, оператор new, сборка мусора. Модификаторы доступа, инициализация объектов. Рекурсия, псевдорекурсия. Статические классы.

7. Методы, использование параметров метода, модификаторы параметров ref и out, использование переменного числа аргументов, перегрузка методов, необязательные аргументы, именованные аргументы, метод Main(), статические методы.

8. Перегрузка операторов: перегрузка унарных и бинарных операторов, операторов отношения, true и false, логических операторов, операторов преобразования.

9. Индексаторы и свойства: одномерные и многомерные индексаторы, перегрузка индексаторов; автосвойства, модификаторы доступа в аксессорах.

10. Наследование: организация защищенного доступа, конструкторы и наследование, сокрытие имен, виртуальные методы, абстрактные классы, ключевое слово sealed, класс object, упаковка и распаковка.

11. Интерфейсы: реализация, применение интерфейсных ссылок, наследование интерфейсов, явные реализации. Структуры и перечисления.

12. Обработка исключительных ситуаций: класс System.Exception, ключевые слова try, catch, throw, finally.

13. Делегаты и события: объявление делегатов, групповая адресация, ковариантность и контрвариантность, анонимные методы, события, групповая адресация события, аксессоры событий.

14. Обобщения: объявление, ограниченные типы, получение значения, присваиваемого параметру типа по умолчанию, обобщенный метод, вариантные делегаты.

15. Этапы разработки программного обеспечения. Общая последовательность, результаты каждого этапа, участники каждого этапа. Виды программной документации.

16. Техническое задание. Назначение документа. Содержание документа и его основных пунктов. Постановка цели и задач. Определение целевой аудитории. Критерии качества продукта.

17. UML-диаграммы. Диаграммы классов. Ассоциация, использование, агрегация, наследование, реализация, обобщение. Кардинальность. Обозначение модификаторов доступа, статических членов класса, виртуальных членов класса на диаграмм классов.

18. UML-диаграммы. Диаграммы активностей. Процессы, передача управления, параллельное выполнение, условия, циклы, суперпроцессы.

Согласована на портале № 48018

19. UML-диаграммы. Диаграммы вариантов использования. Актер, действие, связи: включение, наследование, обобщение, точки обобщения. Назначение и применение диаграмм.

20. Прототипирование. Назначение прототипов. Виды прототипов: горизонтальные и вертикальные, одноразовые и эволюционные, бумажные и электронные.

21. Проектирование интерфейсов. Верстка элементов пользовательского интерфейса.

22. Проектирование интерфейсов. Шаблоны пользовательского поведения: безопасное исследование, мгновенное вознаграждение, разумная достаточность, изменения на полпути, отложенный выбор.

23. Проектирование интерфейсов. Шаблоны пользовательского поведения: пошаговое построение, привыкание, пространственная память, проспективная память, организованное повторение, только клавиатура, советы других людей.

24. Критерии качества спроектированного класса: минимальная сложность, простота сопровождения, слабое сопряжение, расширяемость, возможность повторного использования.

25. Критерии качества спроектированного класса: высокий коэффициент объединения по входу, низкий коэффициент разветвления по выходу, портируемость, достаточная функциональность, стратификация.

26. Оптимизация кода и рефакторинг. Нотация RSDN. Документирование кода.

27. Тестирование. Ошибка, эталон, тест, тестовый случай. Типы тестирования: по знанию внутренней системы, по объекту тестирования.

28. Тестирование. Ошибка, эталон, тест, тестовый случай. Типы тестирования: по степени изолированности тестируемых компонентов, по степени автоматизированности, по степени подготовки к тестированию, по ожидаемому результату.

29. Тестирование. Блочное тестирование: основные принципы, назначение. Способы определения тестовых случаев. Библиотека NUnit. Атрибуты Test, TestCase, Combinatorial, ExpectedException, Repeat, Setup. Класс Assert библиотеки NUnit.

30. Системы версионного контроля. Назначение. Ревизия, репозиторий, рабочая копия. Принципы синхронизации. Конфликты. Ветки и метки.

31. Методологии разработки ПО. Водопадная методология. Преимущества и недостатки. Методология Scrum: общие принципы, виды собраний, основные этапы. Преимущества и недостатки.

32. Методологии разработки ПО. Экстремальное программирование. Основные принципы. Преимущества и недостатки. Методология Kanban, основные принципы, преимущества и недостатки.

14.1.6. Темы лабораторных работ

Проектирование и реализация бизнес-логики приложения
Разработка пользовательского интерфейса приложения на основе WinForms
Работа с системами версионного контроля
Написание автоматизированных модульных тестов
Рефакторинг программы и сборка установщика
Составление проекта системы

Согласована на портале № 48018

14.1.7. Методические рекомендации

не предусмотрены

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;
- в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

- в форме электронного документа;
- в печатной форме.

Согласована на портале № 48018

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

Согласована на портале № 48018