

Министерство образования и молодежной политики Свердловской области
Государственное автономное профессиональное образовательное учреждение
Свердловской области «Екатеринбургский монтажный колледж»

УТВЕРЖДАЮ:

Директор _____ В.Н.Чистяков

« 01 » _____ декабря _____ 2023г



**ПРОГРАММА ПРОФЕССИОНАЛЬНОГО МОДУЛЯ СРЕДНЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
РАЗРАБОТКА И ТЕСТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ**

Категория слушателей: студенты СПО

Уровень квалификации: 4

Объем: 244 академ. часа

Срок: 6 месяцев

Форма обучения: Очная

Организация процесса обучения: с применением ДОТ

Екатеринбург, 2023

Программа профессионального модуля является частью основной профессиональной образовательной программы среднего профессионального образования подготовки специалистов среднего звена по специальности 09.02.07 «Информационные системы и программирование». Программа предусматривает наличие знаний слушателей курса по следующим направлениям:

1. знание основ верстки с использованием языков описания стилей;
 2. знание основ программирования с использованием сценарных языков;
 3. знание методов и приемов алгоритмизации поставленных задач;
 4. знание алгоритмов решения типичных задач, области и способы их применения;
 5. знание методологий разработки компьютерного программного обеспечения и технологий программирования;
 6. знание методологий и технологии проектирования и использования баз данных;
 7. знание методов обработки текстовой, численной и графической информации;
 8. знание принципов создания пользовательских интерфейсов;
 9. умение применять табличные процессоры для обработки числовых данных;
 10. умение применять графические редакторы для создания и обработки изображений;
 11. умение разрабатывать графический дизайн интерфейсов пользователя
- Программа состоит из двух модулей:

1. Разработка кода информационных систем;
2. Тестирование информационных систем

Разработчик(и): Грибова Анна Владимировна Преподаватель

Оглавление

1. Общая характеристика программы	4
1.1. Нормативно-правовые основания разработки программы	4
1.2. Цели реализации программы	4
1.3. Требования к слушателям	5
1.4. Требования к результатам освоения программы	5
2. Учебный план	7
3. Календарный учебный график	8
4. Программы учебных модулей	9
5. Организационно-педагогические условия реализации программы	37
5.1. Материально-техническое обеспечение	37
5.2. Кадровое обеспечение	39
5.3. Организация образовательного процесса	40
5.4. Информационное обеспечение обучения	40
6. Контроль и оценка результатов освоения программы	42
Фонд оценочных средств	44

1. Общая характеристика программы

1.1. Нормативно-правовые основания разработки программы

1. Федеральный закон “Об образовании в Российской Федерации” от 29.12.2012 № 273-ФЗ.
2. Приказ Министерства образования и науки РФ от 1 июля 2013 г. № 499 “Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам”.
3. Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от 09 декабря 2016г. № 1547.
4. Профессиональный стандарт 06.001 «Программист» (утвержден приказом Министерством труда и социальной защиты Российской Федерации от 20 июля 2022 года № 424н.)
5. Профессиональный стандарт «Специалист по тестированию в области информационных технологий» (утвержден приказом Министерства труда и социальной защиты Российской Федерации от 02.08.2021 № 531н);
6. Профессиональный стандарт «Специалист по информационным системам» (утвержден приказом Минтруда России от 18 ноября 2014 года № 896 н).

1.2. Цели реализации программы

Целью реализации программы является освоение вида профессиональной деятельности “Проектирование и разработка информационных систем”, предусмотренным ФГОС по специальности 09.02.07 “Информационные системы и программирование” по квалификации “Разработчик веб и мультимедийных приложений” .

1.3. Требования к слушателям

К освоению программы допускаются:

1. лица, имеющие или получающие среднее профессиональное и (или) высшее образование по направлению Информационные системы,
2. лица, получающие среднее профессиональное образование по укрупненной группе 09.00.00 Информатика и вычислительная техника и освоившие общепрофессиональные дисциплины.
Медицинские ограничения регламентированы Перечнем медицинских противопоказаний Минздрава России.

1.4. Требования к результатам освоения программы

Результатом освоения программы является освоение следующих компетенций:

Код	Наименование профессиональной компетенции
ПК 1.1	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 1.2	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием
ПК 2.1	Разработка тестовых наборов и тестовых сценариев для выполнения тестовых процедур программного обеспечения
ПК 2.2	Тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

Код	Наименование общей компетенции
ОК 1.1	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 1.2	Пользоваться профессиональной документацией на государственном и иностранном языках

ОК 2.1	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 2.2	Планировать и реализовывать собственное профессиональное и личностное развитие

2. Учебный план

Наименование компонентов программы	Объем программы (академические часы)							
	Всего	В том числе с применение м ДОТ и ЭО	Самостоятельн ая работа	Консультац ия	Нагрузка во взаимодействии с преподавателями			
					Теоретическ ое обучение	Практически е и лабораторны е работы	Практика (стажиров ка)	Промежуточна я аттестация, форма
Модуль 1 Разработка кода информационны х систем	140				54	82		4, Экзамен
Модуль 2 Тестирование информационны х систем	98				48	46		4, Зачёт с оценкой
Итоговая аттестация	6							Экзамен
Итого по программе	244							

3. Календарный учебный график

Компоненты программы	Вид учебной нагрузки	Временные параметры (месяцев)																								Всего	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
Модуль 1 Разработка кода информационных систем	Аудиторное обучение	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	4		136	
	Промежуточная аттестация																								4		4
Модуль 2 Тестирование информационных систем	Аудиторное обучение	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	6		94	
	Промежуточная аттестация																								4		4
Итоговая аттестация	Экзамен																								6		6
Итого в неделю		10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	14		244	

4. Программы учебных модулей

4.1. Модуль 1. Разработка кода информационных систем

Модуль направлен на:

1. разработку приложений в соответствии с требованиями технического задания ;
2. написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных;
3. создание графического пользовательского интерфейса по концепции или по образцу уже спроектированной части интерфейса;
4. проверку качества разработанного программного продукта;
5. создание приложений с многоуровневой архитектурой;
6. использование библиотек программных модулей, фреймворков при разработке компьютерного программного обеспечения;
7. использование существующих типовых решений и шаблонов проектирования компьютерного программного обеспечения;
8. Написание программного кода для обеспечения сетевого взаимодействия программных модулей, в том числе разработка процедур интеграции программных модулей;
9. Написание программного кода, обеспечивающего безопасное хранение, обработку и передачу данных;
10. осуществление сборки однородных программных модулей в программный проект;
11. осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта;
12. проверка и отладка программного кода;
13. осуществление обратной связи с пользователем программного продукта на уровне графического пользовательского интерфейса.

4.1.1. Цели реализации модуля

1. Приобретение навыков программирования: освоение основных концепций и технологий программирования, умение разрабатывать эффективные алгоритмы для решения задач и оптимизации работы программы;
2. анализ поставленной задачи и ее реализация средствами языка программирования.
3. Создание приложений в соответствии с техническим заданием на разработку
4. реализация графического пользовательского интерфейса (GUI) в соответствии с техническим заданием

4.1.2. Требования к результатам освоения модуля

Результатом освоения модуля является освоение следующих компетенций:

Код	Наименование профессиональной компетенции
ПК 1.1	Производить разработку модулей информационной системы в соответствии с техническим заданием
ПК 1.2	Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием

Код	Наименование общей компетенции
ОК 1.1	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 1.2	Пользоваться профессиональной документацией на государственном и иностранном языках

В результате освоения модуля слушатель должен:

иметь практический опыт:

1. разработки программных модулей приложения и баз данных в соответствии с

- требованиями технического задания;
2. модификации отдельных модулей информационной системы, рефакторинга и оптимизации программного кода.;
 3. создания сетевого сервера и сетевого клиента

знать:

1. спецификации языка программирования;
2. синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования;
3. принципы создания графического пользовательского интерфейса (GUI);
4. прикладные вопросы программирования и языка сценариев для создания программ;
5. методологии разработки компьютерного программного обеспечения (например, объектно-ориентированных технологий);
6. технологии программирования;
7. особенности выбранной среды программирования и системы управления базами данных;
8. принципы создания файлового ввода-вывода;
9. компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними;
10. методы и средства сборки модулей и компонентов компьютерного программного обеспечения;
11. интерфейсы взаимодействия с внешней средой;
12. интерфейсы взаимодействия внутренних модулей системы (шаблонов проектирования MVC (Model-View-Control), MVVM);
13. методы и средства разработки процедур для развертывания компьютерного программного обеспечения;
14. методы и средства миграции и преобразования данных;

15. методы и приемы отладки программного кода;
16. типы и форматы сообщений об ошибках, предупреждений;
17. технологии точного и постоянного контроля версионностью продукта

уметь:

1. использовать системы управления базами данных для построения, хранения и управления структурами и наборами данных для требуемой системы на основе клиент-серверной архитектуры;
2. использовать подходящие версии программного обеспечения, среды разработки и инструменты, предназначенные для изменения существующего и написания нового исходного кода клиент-серверного программного обеспечения;
3. реализовывать различные типы приложений средствами языка программирования: веб-приложения, настольные приложения, API;
4. определять и интегрировать соответствующие библиотеки и фреймворки в программные решения;
5. строить и обслуживать многоуровневые приложения;
6. осуществлять процедуры миграции и преобразования (конвертации) данных;
7. осуществлять отладку программных решений;
8. использовать языки структурного, объектно-ориентированного программирования и языка сценариев для создания независимых программ;
9. разрабатывать графический интерфейс приложения (GUI);
10. создавать программный код в соответствии с техническим заданием (готовыми спецификациями) ;
11. выполнять оптимизацию программного кода с использованием специализированных программных средств;
12. выполнять процедуры развертывания созданного программного обеспечения, управлять версионностью разработанного программного решения;
13. реализовывать сценарии обработки исключений;

14. использовать возможности имеющейся технической и/или программной архитектуры для написания программного кода;

4.1.3. Программа модуля

Наименование разделов, тем модуля	Содержание обучения по темам, наименование и тематика практических занятий, самостоятельной работы. Вид учебных занятий. Виды выполняемых работ.	Объем часов
1	2	3
Тема "Основные инструменты для создания, исполнения и управления информационной системой"	Содержание: Основные инструменты для создания, исполнения и управления информационной системой	12
	<p><i>Лекция</i></p> <p>Выбор интегрированной среды для создания приложений (IDE). Настройка среды разработки. Спецификация языка программирования. Стандарты кодирования. Исследование программного кода на предмет соответствия стандартам кодирования. Основные инструменты среды для создания, исполнения и управления информационной системой. Оптимизации кода информационной системы с учетом модульности и повторного использования кода. Рефакторинг и оптимизация программного кода. Использование существующего кода в качестве основы для анализа и модификации.</p>	4
	<p><i>Лекция</i></p> <p>Сохранение версий проекта, контроль за версионностью программного продукта. Методы организации работы в команде разработчиков. Формирование репозитория проекта, определение уровня доступа в системе контроля версий</p>	2
	<p><i>Лекция</i></p> <p>Многоуровневая архитектура приложения . Паттерны проектирования программного продукта. MVC (Model-View-Control), MVVM</p>	2

	<i>Лекция</i> Обеспечение кроссплатформенности информационной системы	2
	<i>Лекция</i> Сервисно - ориентированная архитектура приложения	2
	Содержание: Разработка и модификация информационных систем	124
	<i>Лекция</i> Особенности объектно-ориентированных и структурных языков программирования. Разработка сценариев работы приложения с помощью специализированных языков .	2
	<i>Практическое занятие</i> Разработка сценариев работы приложения с помощью специализированного языка программирования. Отладка приложений. Организация обработки исключений.	2
Тема "Разработка и модификация информационных систем"	<i>Практическое занятие</i> Реализация взаимодействия "пользователь-программа". Реализация взаимодействия пользователя с приложением при помощи мыши и функциональных клавиш.	4
	<i>Лекция</i> Интерфейсы пользователя. Требования к интерфейсу пользователя. Принципы создания графического пользовательского интерфейса (GUI).	2
	<i>Лекция</i> Обзор элементов управления и их свойств. Элементы управления содержимым. Применение элементов управления при решении прикладных задач	4
	<i>Практическое занятие</i> Практическая реализация принципов создания графического пользовательского интерфейса (GUI) в	4

приложении. Работа с графикой	
<i>Лекция</i> Средства СУБД в технологиях разработки и управления автоматизированной информационной системой. Использование системы управления базами данных для построения, хранения и управления структурами и наборами данных для требуемой системы на основе клиент-серверной архитектуры.	2
<i>Лекция</i> Подключение к базе данных в приложении (использование фреймворков для взаимодействия с базой данных). Получение данных из базы данных. Технологии доступа к информации в удаленных базах данных. Защита информации	2
<i>Лекция</i> Реализация CRUD- запросов к базе данных в приложении (просмотр, удаление, редактирование и отправка данных в базу данных из приложения).	2
<i>Лекция</i> Реализация алгоритмов поиска данных . Нечеткий поиск.	2
<i>Практическое занятие</i> Реализация функционала приложения. Использование фреймворков при разработке приложения.	4
<i>Практическое занятие</i> Реализация функционала приложения. Реализация CRUD- запросов к базе данных в приложении (просмотр, удаление, редактирование и отправка данных в базу данных из приложения).	4
<i>Практическое занятие</i> Реализация функционала приложения. Защита информации и управление доступом к данным. Хэширование. Captcha. Блокировка экрана. Настройка системы доступа к функциям приложения с помощью	4

ролей	
<i>Практическое занятие</i> Реализация функционала приложения. Работа с файлами и каталогами в приложении. Организация файлового ввода-вывода данных. Работа с разными форматами файлов и структурами данных. Json, Xml, Xls, Csv, и другие. Сериализация и десериализация данных	6
<i>Практическое занятие</i> Реализация функционала приложения. Автоматизированное создание документации в приложении	4
<i>Практическое занятие</i> Реализация функционала приложения. Визуализация цифровых данных (дизайн графиков и диаграмм) для графических пользовательских интерфейсов	4
<i>Практическое занятие</i> Реализация функционала приложения. Разработка и реализация отчетов, необходимых пользователям приложений, с графиками и возможностью вывода на печать.	4
<i>Практическое занятие</i> Реализация функционала приложения. Реализация механизмов обратной связи с пользователем посредством интерфейса.	4
<i>Практическое занятие</i> Развертывание программного обеспечения. Инсталляция, настройка и деинсталляция программного обеспечения	2
<i>Лекция</i> Интерфейсы взаимодействия внутренних модулей системы. Паттерны проектирования. Реализация паттернов MVC (Model-View-Control), MVVM	8

	<i>Лекция</i> Проектирование серверной части многопользовательских информационных систем. Общие принципы проектирования клиентской части автоматизированной информационной системы	8
	<i>Практическое занятие</i> Создание сетевого сервера и сетевого клиента	8
	<i>Лекция</i> Разработка и публикация API.	2
	<i>Практическое занятие</i> Разработка и публикация API.	2
	<i>Лекция</i> Создание веб-приложения средствами языка программирования. Реализация паттерна MVC в веб-приложении. Создание контроллеров и представлений. Публикация приложения	8
	<i>Практическое занятие</i> Создание веб-приложения средствами языка программирования. Реализация паттерна MVC в веб-приложении. Создание контроллеров и представлений. Публикация приложения	26
Промежуточная аттестация	Экзамен Разработка информационной системы	4
Итого:		140

4.1.4. Материально-техническое обеспечение

Кабинет (лаборатория), мастерская	Оборудование и технические средства обучения
Мастерская ОМ Программные решения для бизнеса, ЦПДЭ	1 WiFi-точка доступа 2 аптечка

- 3 Интерфейсный кабель для подключения монитора
 - 4 Кабель питания
 - 5 Клавиатура
 - 6 Компьютерная мышь
 - 7 Компьютерный монитор
 - 8 Компьютерный стул
 - 9 Кондиционер
 - 10 Маршрутизатор
 - 11 Ноутбук
 - 12 Огнетушитель углекислотный ОУ-1
 - 13 Офисный стол
 - 14 Персональный компьютер в сборе
 - 15 Планшеты на Android
 - 16 Проектор
 - 17 сервер
 - 18 Сетевой фильтр
 - 19 Управляемый коммутатор
 - 20 Экран для проектора
-
- 1 Веб-браузер Chrome
 - 2 Веб-браузер Opera
 - 3 ПО Anaconda
 - 4 ПО Android Studio
 - 5 ПО API Postman
 - 6 ПО Eclipse IDE for Java Developers
 - 7 ПО e(fx)clipse
 - 8 ПО Hibernate ORM
 - 9 ПО IntelliJ IDEA

- 10 ПО Java SE Development Kit
- 11 ПО Microsoft JDBC Driver for SQL Server
- 12 ПО Microsoft SQL Server Express
- 13 ПО Microsoft Visual Studio Code
- 14 ПО MySQL Installer
- 15 ПО NetBeans
- 16 ПО .NET Framework Developer pack
- 17 ПО NGINX
- 18 ПО Notepad++
- 19 ПО PyCharm
- 20 ПО Python
- 21 ПО SQLAlchemy
- 22 ПО SQL Server Management Studio
- 23 ПО администрирования DBeaver
- 24 ПО для архивации WinRAR
- 25 ПО для просмотра документов в формате PDF Adobe Reader DC
- 26 ПО для управления версиями Gogs
- 27 ПО операционная система Microsoft Windows 10 Pro
- 28 ПО офисный пакет Microsoft Office 2019
- 29 ПО СУБД POSTGRESQL
- 30 ПО текстовый редактор
- 31 ПО управления версиями Git
- 32 ПО Эмулятор Android
- 33 Программная платформа .NET 5
- 34 Программное обеспечение 7-Zip
- 35 Программное обеспечение dbForge Studio for PostgreSQL
- 36 Программное обеспечение MySQL Installer Community ,
включая следующие компоненты: MySQL Workbench;
MySQL for Visual Studio; Connector/NET; Connector/ODBC;

Connector/J; Connector/Python.

37 Программное обеспечение серверная операционная система с интегрированной системой виртуализации Hyper-V
--

4.1.5. Кадровое обеспечение

Реализация образовательной программы обеспечивается педагогическими работниками колледжа, направление деятельности которых соответствует области профессиональной деятельности укрупненной группы специальностей 09.00.00 “Информатика и вычислительная техника” . Педагогические работники, привлекаемые к реализации образовательной программы, получают дополнительное профессиональное образование по программам повышения квалификации, в том числе в форме стажировки в организациях, направление деятельности которых соответствует направлению обучения по специальностям укрупненной группы 09.00.00 “Информатика и вычислительная техника” не реже 1 раза в 3 года с учетом расширения спектра профессиональных компетенций.

4.1.6. Организация образовательного процесса

Организация образовательного процесса регламентируется учебным планом и расписанием учебных занятий. Язык обучения (преподавания) - русский. Устанавливаются следующие формы учебных занятий: практическое занятие, лекция. Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут. Для проведения практических занятий слушателям предлагаются подробные инструкции по выполнению. Обучение ведется с использованием системы дистанционного обучения ГАПОУ СО “ЕМК”. Организация образовательного процесса предусматривает применение активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой, а так же использование дистанционной формы обучения. Изучение программы модуля завершается экзаменом с оценкой (фонд оценочных средств модуля представлен в приложении 2).

Изучение программы модуля завершается дифференцированным зачетом с оценкой (зачетные задания представлены в Приложении 1). Слушатели выполняют итоговое задание, направленное на демонстрацию освоения принципов разработки приложений.

4.1.7. Информационное обеспечение обучения

Основная литература:

1. Лисяк, В. В. Разработка информационных систем : учебное пособие / В. В. Лисяк ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2019. - 96 с.

Дополнительная литература:

1. Шитов, В. Н. Разработка информационного контента (по отраслям) : учебное пособие / В.Н. Шитов. — Москва : ИНФРА-М, 2022. — 178 с.

2. Кузниченко, М. А. Основы баз данных : учебно-методическое пособие / М. А. Кузниченко. - 2-е изд., стер. - Москва : Флинта, 2022. - 102 с. - ISBN 978-5-9765-5139-8.

4.1.8. Контроль и оценка результатов освоения модуля

Результат освоения программы	Основные показатели оценки результата
ПК 1.1 Производить разработку модулей информационной системы в соответствии с техническим заданием	Разработка модулей информационной системы выполнена в соответствии с техническим заданием
ПК 1.2 Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием	Разработка подсистем безопасности информационной системы выполнена в соответствии с техническим заданием
ОК 1.1 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Использование источников информации по профилю специальности

ОК 1.2 Пользоваться профессиональной документацией на государственном и иностранном языках	Использование технической документации в работе
--	---

Форма и вид аттестации по модулю:

Форма аттестации по модулю - дифференцированный зачет. Фонд оценочных средств представлен в Приложении 1. Задания для проведения экзамена направлены на:

1. разработку модулей клиент-серверного приложения;
2. организацию CRUD-запросов к базе данных;
3. обработку ошибок

4.2. Модуль 2. Тестирование информационных систем

Тестирование программного продукта тесно связано с его разработкой. Задача тестирования состоит в поиске ошибок, отказов и дефектов в приложениях, валидации и верификации программного продукта. В процессе изучения модуля, слушатели изучат виды и уровни тестирования, создадут и оформят тестовые сценарии для тестирования созданных программных модулей, научатся создавать авто- и юнит-тесты.

Модуль направлен на

1. обеспечение качества программных продуктов;
2. подготовку тестовых наборов данных в соответствии с выбранной методикой тестирования компьютерного программного обеспечения;
3. проверку работоспособности компьютерного программного обеспечения на основе разработанных тестовых наборов данных;
4. оценку соответствия компьютерного программного обеспечения требуемым характеристикам;

5. сбор и анализ полученных результатов проверки работоспособности компьютерного программного обеспечения;
 6. сбор диагностических данных проверки работоспособности компьютерного программного обеспечения.
 7. проверка соответствия действительных и указанных в эксплуатационной и технической документации на ПО результатов;
 8. при выявлении несоответствий действительных и указанных в эксплуатационной и технической документации результатов регистрация найденных дефектов ПО в системе контроля дефектов;
 9. выявление приоритетных областей покрытия тестовыми случаями на основе плана тестирования ПО ;
 10. описание тестовых случаев;
 11. выделение классов эквивалентности значений каждого типа входных данных;
 12. составление списка комбинаций значений из различных классов эквивалентности;
 13. построение тестовых случаев, в которых сочетаются одна перестановка значений с необходимыми внешними ограничениями;
 14. написание/настройка программ для автоматизированного тестирования ПО (при необходимости);
 15. разработка автоматизированных тестов, в том числе для проверки информационной безопасности разрабатываемого ПО;
 16. инструктирование специалистов по подготовке тестовых данных и выполнению тестовых процедур ПО;
 17. тестирование разрабатываемого модуля ИС;
 18. устранение обнаруженных несоответствий в модуле ИС;
 19. интеграционное тестирование ИС на основе тест-планов;
 20. фиксирование результатов тестирования в системе учета.
- Модуль включает большое количество практических заданий, направленных

на освоение различных методов тестирования программного обеспечения, написание автоматизированных тестов для проверки работоспособности программ.

4.2.1. Цели реализации модуля

Цели реализации модуля:

1. Знакомство с профессией тестировщика программного обеспечения;
2. Составлять документацию : оформлять документацию: тест-кейсы, чек-листы, тест-планы, описывать дефекты и баги для разработчиков
3. Разрабатывать автотесты любой сложности;
4. Проводить комплексное ручное тестирование ПО.

4.2.2. Требования к результатам освоения модуля

Результатом освоения модуля является освоение следующих компетенций:

Код	Наименование профессиональной компетенции
ПК 2.1	Разработка тестовых наборов и тестовых сценариев для выполнения тестовых процедур программного обеспечения
ПК 2.2	Тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы

Код	Наименование общей компетенции
ОК 2.1	Пользоваться профессиональной документацией на государственном и иностранном языках
ОК 2.2	Планировать и реализовывать собственное профессиональное и личностное развитие

В результате освоения модуля слушатель должен:

иметь практический опыт:

1. использования критериев оценки качества и надежности информационной системы;
2. применения методик тестирования разрабатываемых приложений;
3. проведения тестирования в соответствии с заданием;
4. проведения интеграционного тестирования ИС на основе тест-планов в соответствии с трудовым заданием;
5. документирования результатов тестов;
6. устранения обнаруженных несоответствий;
7. фиксирования результатов тестирования в системе учета.

знать:

1. языки, утилиты и среды программирования и средства пакетного выполнения тестовых процедур;
2. методы контроля качества программного продукта в соответствии со стандартами;
3. основы верификации и аттестации программного обеспечения (ПО);
4. особенности программных средств, используемых в разработке и тестировании информационной системы;
5. устройство и функционирование современных информационных систем;
6. стандарты качества программного продукта и процессов его обеспечения;
7. роль тестирования в жизненном цикле ПО;
8. основные понятия и принципы тестирования ПО;
9. основные понятия автоматизации тестирования программного обеспечения;
10. техники тест-дизайна;
11. базовые принципы взаимодействия с REST API, типичные сценарии тестирования API-интерфейсов;
12. инструменты и методы модульного тестирования;

13. современные методики тестирования разрабатываемых информационных систем;
14. методы создания и документирования контрольных примеров и тестовых наборов данных;
15. правила, алгоритмы и технологии создания тестовых наборов данных;
16. требования к структуре и форматам хранения тестовых наборов данных;
17. методы и средства проверки работоспособности компьютерного программного обеспечения;
18. среды проверки работоспособности и отладки компьютерного программного обеспечения;
19. методы автоматической и автоматизированной проверки работоспособности компьютерного программного обеспечения;
20. типовые метрики компьютерного программного обеспечения и основные методы измерения и оценки характеристик компьютерного программного обеспечения;
21. основные понятия о качестве ПО;
22. выявление приоритетных областей покрытия тестовыми случаями на основе плана тестирования ПО;
23. идентификация всех значений, которые вводятся участниками в сценарии использования системы;
24. выделение классов эквивалентности значений каждого типа входных данных;
25. составление списка комбинаций значений из различных классов эквивалентности;
26. построение тестовых случаев, в которых сочетаются одна перестановка значений с необходимыми внешними ограничениями;
27. написание/настройка программ для автоматизированного тестирования ПО (при необходимости) .

уметь:

1. Планировать собственную работу в рамках полученного задания в зависимости от стадии тестирования;
2. Определять функциональные и нефункциональные требования спецификации;
3. Анализировать требования с целью выявления их недостатков;
4. Использовать методы тестирования в соответствии с техническим заданием;
5. Тестировать результаты собственной работы;
6. Выполнять тестовые процедуры по тестированию информационной системы согласно тест-плану;
7. Использовать готовый тест-план для планирования собственной работы ;
8. Использовать техники и подходы к проектированию тестовых испытаний;
9. Декомпозировать тестируемые объекты и поставленные задачи;
10. Разрабатывать тест-кейсы и проверять результаты тест-кейсов;
11. Анализировать готовые тест-кейсы, обнаруживать и устранять имеющиеся в них недостатки;
12. Составлять отчет о процессе тестирования (выполнять документирование тестовых испытаний)
13. Проводить тестирование интерфейса пользователя на основе готовых тестовых сценариев;
14. Предоставлять необходимую информацию для формирования отчёта о результатах тестирования;
15. Анализировать готовые отчёты о результатах тестирования с целью уточнения планирования собственной работы;
16. Разрабатывать требования к системе, тесты и тестовые процедуры и отслеживать взаимосвязь этих документов с разработанными тестами;
17. Разрабатывать различные виды тестов и тестирующих программ;
18. Разрабатывать инструменты для автоматизированного тестирования программного обеспечения.

4.2.3. Программа модуля

Наименование разделов, тем модуля	Содержание обучения по темам, наименование и тематика практических занятий, самостоятельной работы. Вид учебных занятий. Виды выполняемых работ.	Объем часов
1	2	3
Тема "Основы тестирования программного обеспечения"	Содержание: Основы тестирования программного обеспечения	16
	<p><i>Лекция</i></p> <p>Метрики и атрибуты оценки качества программного обеспечения. Анализ спецификаций . Понятие процесса тестирования программного обеспечения. Этапы процесса тестирования ПО. Основные определения тестирования. Цели и задачи процесса тестирования. Полный цикл тестирования. (фазы тестирования, процесс тестирования на разных этапах разработки ПО). Валидация. Верификация.</p>	4
	<p><i>Лекция</i></p> <p>Методы и виды тестирования. Классификация видов тестирования. Объекты тестирования. Базовые инструменты тестировщика. Организация тестирования в команде разработчиков. Современные методики тестирования разрабатываемых ИС.</p>	4
	<p><i>Лекция</i></p> <p>Стратегии тестирования. Стратегии тестирования: черного ящика, белого ящика. Тестирования с применением функциональных диаграмм. Тестирования методом предположения об ошибке. Предположение ошибок. Применение техник тестирования для разработки тест-кейсов с учетом особенностей физической и логической структуры</p>	4
	<p><i>Лекция</i></p> <p>Понятие программной ошибки. Анализ ошибки. Модель работы с дефектами. Жизненный цикл бага. Ошибки программного обеспечения: источники ошибок ПО, классификация ошибок, основные пути борьбы с ошибками. Критерии покрытия кода</p>	2

	программы тестами.	
	<i>Практическое занятие</i> Определение объектов тестирования. Формирование набора тестовых случаев на примере конкретного программного продукта. Создание чек-листа	2
Тема "Методы и принципы тестирования программного обеспечения"	Содержание: Ручное тестирование программного обеспечения	14
	<i>Лекция</i> Инструменты тестирования. Позитивные и негативные тест-кейсы. Техники тестирования: Boundary Values Analysis (Анализ граничных значений) и Equivalence Partitioning (Разделение на эквивалентные классы). Комбинирование параметров тест-кейса при доменном тестировании.	4
	<i>Практическое занятие</i> Тестовая документация. Виды документации. Тест-план, чекк-лист, тест-кейс	2
	<i>Лекция</i> Техники тестирования: Попарное тестирование (Pairwise testing) и поиск комбинаций, исследовательское тестирование (Exploratory testing). Поиск причин возникновения дефектов	2
	<i>Практическое занятие</i> Тестирование настольного приложения. Определение объектов тестирования, формирование чек-листа тестирования, оформление тест-кейсов, оформление отчетов об ошибках.	2
	<i>Практическое занятие</i> Тестирование веб-сервиса. Определение объектов тестирования, формирование чек-листа тестирования, оформление тест-кейсов, оформление отчетов об ошибках.	2
	<i>Практическое занятие</i> Тестирование мобильных приложений. Определение	2

	объектов тестирования, формирование чек-листа тестирования, оформление тест-кейсов, оформление отчетов об ошибках	
Тема "Unit-тестирование"	Содержание: Unit-тестирование	32
	<i>Лекция</i>	4
	Процесс модульного тестирования, его задачи и цели. Понятие модуля и его границ. Тестовое окружение при модульном тестировании. Test Driven Development (разработка через тестирование). Фреймворки для unit-тестирования. Оформление unit-теста	
	<i>Практическое занятие</i>	4
	Разработка тестового сценария проекта. Разработка unit-тестов.	
	<i>Практическое занятие</i>	4
	Обработка исключительных ситуаций в unit-тестах. Документирование процесса тестирования	
	<i>Практическое занятие</i>	6
Функциональное тестирование. Разработка unit-тестов. Документирование процесса тестирования. Test Driven Development		
<i>Лекция</i>	2	
Интеграционное тестирование. Способы интеграционного тестирования. Системное тестирование.		
<i>Практическое занятие</i>	4	
Тестирование интеграции. Разработка интеграционных тестов в IDE. Создание и документирование unit-тестов интеграционного тестирования		
<i>Практическое занятие</i>	8	
Создание unit-тестов для проверки работоспособности веб-приложении. Документирование процесса тестирования		

Тема "Автоматизация тестирования программного обеспечения"	Содержание: Автоматизация тестирования программного обеспечения	32
	<i>Лекция</i> Автоматизация тестирования. Сокращение тестовых наборов. Автоматизация приемочного и регрессионного тестирования. Основные направления автоматизации тестирования. Автоматизация построения тестов, символьное исполнение программ .	2
	<i>Лекция</i> Инструменты автоматизации тестирования. Определение направлений автоматизации. Сравнительный анализ инструментов автоматизации	2
	<i>Лекция</i> Автоматизация тестирования при использовании различных интерфейсов (API, CLI, GUI и другие). Нефункциональное тестирование: тестирование производительности, безопасности, удобства использования. Документирование процесса тестирования	6
	<i>Практическое занятие</i> Автоматизация процесса тестирования. Нагрузочное тестирование. Документирование процесса тестирования	2
	<i>Практическое занятие</i> Автоматизация процесса тестирования. Стрессовое тестирование. Документирование процесса тестирования	2
	<i>Практическое занятие</i> Регрессионное тестирование. Использование инструментария анализа качества программного средства. Разработка баг-репорта	2
	<i>Лекция</i> Автоматизация тестирования. Построение минимального дугового покрытия управляющего графа программы и на его основе минимального	4

	набора тестов для критерия С1. Контроль качества набора тестов. Системы контроля полноты набора тестов для определенных критериев. Взаимодействие автотестов с тестируемой системой.	
	<i>Лекция</i> Подходы к автоматизации Функциональная декомпозиция. Data-driven. Keyword-driven. Behavior-driven.	2
	<i>Лекция</i> Проект Selenium. Поддерживаемые платформы. Компоненты Selenium Webdriver. Взаимодействие с браузером и UI	6
	<i>Практическое занятие</i> Работа с Selenium. Установка системы. Создание автотеста.	4
Промежуточная аттестация	Зачёт с оценкой Дифференцированный зачет	4
Итого:		98

4.2.4. Материально-техническое обеспечение

Кабинет (лаборатория), мастерская	Оборудование и технические средства обучения
Мастерская ОМ Программные решения для бизнеса, ЦПДЭ	1 WiFi-точка доступа
	2 аптечка
	3 Интерфейсный кабель для подключения монитора
	4 Кабель питания
	5 Клавиатура
	6 Компьютерная мышь
	7 Компьютерный монитор
	8 Компьютерный стул
	9 Кондиционер

10	Маршрутизатор
11	Ноутбук
12	Огнетушитель углекислотный ОУ-1
13	Офисный стол
14	Персональный компьютер в сборе
15	Планшеты на Android
16	Проектор
17	сервер
18	Сетевой фильтр
19	Управляемый коммутатор
20	Экран для проектора
1	Веб-браузер Chrome
2	Веб-браузер Opera
3	ПО API Postman
4	ПО IntelliJ IDEA
5	ПО администрирования DBeaver
6	ПО для архивации WinRAR
7	ПО для просмотра документов в формате PDF Adobe Reader DC
8	ПО операционная система Microsoft Windows 10 Pro
9	ПО офисный пакет Microsoft Office 2019
10	ПО управления версиями Git

4.2.5. Кадровое обеспечение

Реализация образовательной программы обеспечивается педагогическими работниками колледжа, направление деятельности которых соответствует области профессиональной деятельности укрупненной группы специальностей 09.00.00 “Информатика и вычислительная техника” . Педагогические работники,

привлекаемые к реализации образовательной программы, получают дополнительное профессиональное образование по программам повышения квалификации, в том числе в форме стажировки в организациях, направление деятельности которых соответствует направлению обучения по специальностям укрупненной группы 09.00.00 “Информатика и вычислительная техника” не реже 1 раза в 3 года с учетом расширения спектра профессиональных компетенций.

4.2.6. Организация образовательного процесса

Организация образовательного процесса регламентируется учебным планом и расписанием учебных занятий. Язык обучения (преподавания) - русский. Устанавливаются следующие формы учебных занятий: практическое занятие, лекция. Для всех видов аудиторных занятий академический час устанавливается продолжительностью 45 минут.

Организация образовательного процесса предусматривает применение активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой, а так же использование дистанционной формы обучения. Изучение программы модуля завершается экзаменом с оценкой (фонд оценочных средств модуля представлен в приложении 2).

4.2.7. Информационное обеспечение обучения

Основная литература:

1. Орлов, С. А. Технологии разработки программного обеспечения : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - Санкт-Петербург : Питер, 2021. - 608 с.

2. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспечения.- М: Издательство: Инфра-М, 2022

3. Галина Иванова. Технология программирования. Проектирование. Комплексное тестирование. Программная документация. Издательство: Кнорус, 2022 г.

Дополнительная литература:

1. Морозова, Ю. В. Тестирование программного обеспечения : учебное пособие / Ю. В. Морозова. - Томск : Эль-Контент, 2019.

Электронные и интернет-ресурсы:

1. Тестирование программного обеспечения. Базовый курс [Электронный ресурс] URL: https://svyatoslav.biz/software_testing_book/ (дата обращения: 21.02.2024г.).

4.2.8. Контроль и оценка результатов освоения модуля

Результат освоения программы	Основные показатели оценки результата
ПК 2.1 Разработка тестовых наборов и тестовых сценариев для выполнение тестовых процедур программного обеспечения	Составление тест-плана для выполнение тестовых процедур
ПК 2.2 Тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы	Качество проведенного исследования заявленной в требованиях функциональности приложения. Качество тестовой документации, создание отчетов по тестированию
ОК 2.1 Пользоваться профессиональной документацией на государственном и иностранном языках	Использование в работе профессиональной документации
ОК 2.2 Планировать и реализовывать собственное профессиональное и личностное развитие	Планирование и реализация практических задач, исследовательская деятельность

Форма и вид аттестации по модулю:

Форма аттестации по модулю - дифференцированный зачет. Фонд оценочных средств представлен в Приложении 2. Задания для проведения экзамена направлены на:

1. создание библиотеки классов;

2. создание unit-тестов, проверяющих корректность работы реализованных методов;
3. документирование тест-кейсов

5. Организационно-педагогические условия реализации программы

5.1. Материально-техническое обеспечение

Кабинет (лаборатория), мастерская	Оборудование и технические средства обучения
Мастерская ОМ Программные решения для бизнеса, ЦПДЭ	1 WiFi-точка доступа 2 аптечка 3 Интерфейсный кабель для подключения монитора 4 Кабель питания 5 Клавиатура 6 Компьютерная мышь 7 Компьютерный монитор 8 Компьютерный стул 9 Кондиционер 10 Маршрутизатор 11 Ноутбук 12 Огнетушитель углекислотный ОУ-1 13 Офисный стол 14 Персональный компьютер в сборе 15 Планшеты на Android 16 Проектор 17 сервер 18 Сетевой фильтр 19 Управляемый коммутатор 20 Экран для проектора 1 Веб-браузер Chrome 2 Веб-браузер Opera

- 3 ПО Anaconda
- 4 ПО Android Studio
- 5 ПО API Postman
- 6 ПО Eclipse IDE for Java Developers
- 7 ПО e(fx)clipse
- 8 ПО Hibernate ORM
- 9 ПО IntelliJ IDEA
- 10 ПО Java SE Development Kit
- 11 ПО Microsoft JDBC Driver for SQL Server
- 12 ПО Microsoft SQL Server Express
- 13 ПО Microsoft Visual Studio Code
- 14 ПО MySQL Installer
- 15 ПО NetBeans
- 16 ПО .NET Framework Developer pack
- 17 ПО NGINX
- 18 ПО Notepad++
- 19 ПО PyCharm
- 20 ПО Python
- 21 ПО SQLAlchemy
- 22 ПО SQL Server Management Studio
- 23 ПО администрирования DBeaver
- 24 ПО для архивации WinRAR
- 25 ПО для просмотра документов в формате PDF Adobe Reader DC
- 26 ПО для управления версиями Gogs
- 27 ПО операционная система Microsoft Windows 10 Pro
- 28 ПО офисный пакет Microsoft Office 2019
- 29 ПО СУБД POSTGRESQL
- 30 ПО текстовый редактор

31	ПО управления версиями Git
32	ПО Эмулятор Android
33	Программная платформа .NET 5
34	Программное обеспечение 7-Zip
35	Программное обеспечение dbForge Studio for PostgreSQL
36	Программное обеспечение MySQL Installer Community , включая следующие компоненты: MySQL Workbench; MySQL for Visual Studio; Connector/NET; Connector/ODBC; Connector/J; Connector/Python.
37	Программное обеспечение серверная операционная система с интегрированной системой виртуализации Hyper- V

5.2. Кадровое обеспечение

Реализация образовательной программы обеспечивается педагогическими работниками колледжа, направление деятельности которых соответствует области профессиональной деятельности укрупненной группы специальностей 09.00.00 “Информатика и вычислительная техника” . Педагогические работники, привлекаемые к реализации образовательной программы, получают дополнительное профессиональное образование по программам повышения квалификации, в том числе в форме стажировки в организациях, направление деятельности которых соответствует направлению обучения по специальностям укрупненной группы 09.00.00 “Информатика и вычислительная техника” не реже 1 раза в 3 года с учетом расширения спектра профессиональных компетенций. В подготовке специалистов участвуют преподаватели, имеющие ВКК, 1КК. Количество преподавателей, реализующих обучение по программе, с опытом работы свыше 18 лет - 2 человека.

Доля педагогов, имеющих высшее образование, соответствующее профилю преподаваемой дисциплины (модуля), составляет 100% . 100% педагогов прошли стажировку в профильных организациях. Информация о персональном составе педагогических работников размещена на сайте колледжа.

5.3. Организация образовательного процесса

Обучение проводится в очной форме. Сопровождение обучения ведется с помощью системы дистанционного обучения . Курс системы дистанционного обучения включает подробное описание выполнения практических заданий, видеоматериал по всем изучаемым темам. Курс представляет собой тематически завершённый, структурированный авторами учебный материал, который предназначен для организации самостоятельной работы и используется в качестве веб-поддержки курса.

5.4. Информационное обеспечение обучения

Основная литература:

1. Лисяк, В. В. Разработка информационных систем : учебное пособие / В. В. Лисяк ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2019. - 96 с.

2. Орлов, С. А. Технологии разработки программного обеспечения : учебник для вузов / С. А. Орлов, Б. Я. Цилькер. - 4-е изд. - Санкт-Петербург : Питер, 2021. - 608 с.

3. Гагарина Л.Г., Кокорева Е.В., Сидорова-Виснадул Б.Д. Технология разработки программного обеспечения.- М: Издательство: Инфра-М, 2022

4. Галина Иванова. Технология программирования. Проектирование. Комплексное тестирование. Программная документация. Издательство: Кнорус, 2022 г.

Дополнительная литература:

1. Шитов, В. Н. Разработка информационного контента (по отраслям) : учебное пособие / В.Н. Шитов. — Москва : ИНФРА-М, 2022. — 178 с.

2. Кузниченко, М. А. Основы баз данных : учебно-методическое пособие / М. А. Кузниченко. - 2-е изд., стер. - Москва : Флинта, 2022. - 102 с. - ISBN 978-5-9765-5139-8.

3. Морозова, Ю. В. Тестирование программного обеспечения : учебное пособие / Ю. В. Морозова. - Томск : Эль-Контент, 2019.

Электронные и интернет-ресурсы:

1. Тестирование программного обеспечения. Базовый курс [Электронный ресурс] URL: https://svyatoslav.biz/software_testing_book/ (дата обращения: 21.02.2024г.).

6. Контроль и оценка результатов освоения программы

Результат освоения программы	Основные показатели оценки результата
ПК 1.1 Производить разработку модулей информационной системы в соответствии с техническим заданием	Разработка модулей информационной системы выполнена в соответствии с техническим заданием
ПК 1.2 Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием	Разработка подсистем безопасности информационной системы выполнена в соответствии с техническим заданием
ОК 1.1 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	Использование источников информации по профилю специальности
ОК 1.2 Пользоваться профессиональной документацией на государственном и иностранном языках	Использование технической документации в работе
ПК 2.1 Разработка тестовых наборов и тестовых сценариев для выполнения тестовых процедур программного обеспечения	Составление тест-плана для выполнения тестовых процедур
ПК 2.2 Тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы	Качество проведенного исследования заявленной в требованиях функциональности приложения. Качество тестовой документации, создание отчетов по тестированию
ОК 2.1 Пользоваться профессиональной документацией на государственном и иностранном языках	Использование в работе профессиональной документации
ОК 2.2 Планировать и реализовывать собственное профессиональное и личностное развитие	Планирование и реализация практических задач, исследовательская деятельность

Контроль и оценка результатов освоения программы:

Оценка качества освоения программы включает итоговую аттестацию слушателей. Форма итоговой аттестации - экзамен. К итоговой аттестации допускаются слушатели, успешно прошедшие промежуточную аттестацию по модулям, предусмотренными учебным планом настоящей программы. Задание для проведения экзамена представлено в Приложении 3.

Итоговая аттестация по программе: Экзамен, экзамен.

Фонд оценочных средств

Приложение №1

Фонд оценочных средств
для проведения аттестации
по модулю: «Разработка информационных систем»

г.Екатеринбург, 2023

Комплект оценочных средств

УСЛОВИЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1. Место выполнения задания: мастерская «Программные решения для бизнеса»
2. Максимальное время выполнения задания: 4 академических часа

ЗАДАНИЕ

В одном из театров города X, некий таинственный путешественник дает представления, где демонстрирует искусство черной магии. Каждое из таких представлений дается бесплатно и собирает полный зал. Поэтому, чтобы упростить процедуру регистрации на мероприятия, необходимо создать приложение, которое позволяет посетителям выполнить регистрацию самостоятельно.

Предполагается, что для *регистрации* посетителю нужно указать:

- фамилию;
- дату рождения;
- номер телефона;
- представление, на которое зритель бы хотел попасть;

О каждом *представлении*, в свою очередь, должно быть известно следующее:

- название представления;
- максимальное количество мест;
- количество свободных мест (не задано в таблице);
- дата и время проведения;
- краткое описание;

Таблица 1. Информация о *представлениях*

1	Языки пламени	50	18.06.2022	Встречайте первый сеанс темной магии! Вы удивитесь тем
---	---------------	----	------------	--

			19:00	чудесам, которые Маэстро выполняет на сцене! Первое представление - демонстрация стихии огня. Огонь из ничего, огонь из всего, огонь во все! Посетите данное представление, и вы узнаете, что огонь может приобретать самые причудливые формы.
2	Золотая лихорадка	35	15.09.2022 20:00	Второй сеанс темной магии обещает вам сказочные богатства! Только 15 сентября, вы сможете с помощью темной магии Маэстро, превратить любые предметы в золото! Не пропустите! Также, представление обещает несколько сцен с исчезновением, и пару магических трюков с перемещением предметов силой мысли!
3	Добро пожаловать на сеанс!	50	30.10.2022 20:00	Встречайте третий сеанс темной магии! В этот раз, вы сможете попросить переместить в зал любого человека и с помощью темной магии этот человек попадет на сцену! Пусть он убедится в серьезности темной магии!

Руководство театра ожидает, что ваша разработка:

- позволит просматривать информацию о *предстоящих* мероприятиях в удобном виде. Если представление уже прошло, то его не должно быть видно;
- позволяет зарегистрировать гостя на интересующее представление (один человек может зарегистрироваться дважды, указав один и тот же номер телефона. Руководство театра надеется на порядочность посетителей и

предполагает, что этого не произойдет). После регистрации гость получает секретный код (в качестве секретного кода может быть использован код регистрации, либо сгенерированный любым другим образом код);

- позволяет отменить участие по указанному коду (код выдается при регистрации). Перед отменой участия, необходимо для проверки показать информацию о посетителе;

При разработке стоит учесть следующее:

- невозможно зарегистрироваться на мероприятие, если нет свободных мест;
- гости не могут изменять информацию о мероприятиях и других гостях;
- будет нехорошо, если например, программа завершится с ошибкой при попытке регистрации пользователя с пустыми полями;
- темная магия - вещь серьезная. Лицам, не достигшим 18 лет в регистрации должно быть отказано.

Фонд оценочных средств

для проведения аттестации

по модулю: «Тестирование информационных систем»

Екатеринбург, 2023

Комплект оценочных средств

Задание направлено на:

1. создание библиотеки классов
2. создание методов в соответствии с заданием (задания для создания методов предлагаются разной сложности . смотрите приложение)
3. создание тестовых наборов и реализацию модульного тестирования методов

КРИТЕРИИ ВЫСТАВЛЕНИЯ ОЦЕНОК

Результаты контроля определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Количество баллов	Оценка
10-14	Отлично
7-9	Хорошо
5-6	Удовлетворительно
3 и ниже	Неудовлетворительно

КРИТЕРИИ ОЦЕНИВАНИЯ

1. Метод реализован (0 - метод отсутствует; 1 - добавлена сигнатура метода; 2,3 - имеется частичное решение; 4 - задача решена полностью);
2. Создан корректный набор тестовых сценариев (0 - тесты отсутствуют; 1,2 - имеется набор тестов, однако есть определенные недочеты; 3 - набор тестов составлен правильно);

3. Создан документация к тест-кейсам (0 – документация отсутствует; 1,2 - имеется документация, однако есть определенные недочеты; 3 – документация выполнена по стандартам);

4. Метод имеет XML-комментарий (0 - комментарий отсутствует, либо не информативен; 1 - комментарий составлен как надо);

5. Общая оценка "читаемости" предоставленного кода: форматирование кода, имена переменных, использование подпрограмм и т. п. (0 – код не читаем, либо неинформативен; 1 – код «читаем»);

6. Общая оценка рациональности выбранных методов решений (0 – выбранные способы решения не рациональны; 1, 2 – выбранные способы решения рациональны)

Инструкция по выполнению заданий

- Время для выполнения задания – 4 часа.

ЗАДАНИЕ №1

1. Создать библиотеку классов, в которой реализован метод, выполняющий разворот строки.

Сигнатура метода

```
public static string ReverseString(string textString)
```

Входное значение: в качестве параметра передается строка, содержащая ноль или более символов.

Возвращаемое значение: Возвращается “перевернутая” строка, в которой последний сим-вол станет первым, предпоследний вторым и т. д.

2. Реализовать unit-тестирование созданной библиотеки классов

3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №2

1. Создать библиотеку классов, в которой реализован метод `OnlyLetters`, который бы определял, что строка состоит только лишь из букв.

Сигнатура метода

```
public static bool OnlyLetters(string textString)
```

Входное значение: метод принимает в качестве параметра строку. Предполагается, что строка может быть пустой или содержать произвольный набор символов.

Возвращаемое значение: метод возвращает `true`, если входная строка содержит только буквы. Если имеется хотя бы один символ, которой не является буквой, то метод должен вернуть `false`. В случае пустой строки метод должен вернуть `false`.

2. Реализовать unit-тестирование созданной библиотеки классов

3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №3

1. Создать библиотеку классов, в которой реализован метод `OnlyDigits`, который бы определял, что строка состоит только лишь из цифр.

Сигнатура метода

```
public static bool OnlyDigits(string textString)
```

Входное значение: метод принимает в качестве параметра строку. Предполагается, что строка может быть пустой или содержать произвольный набор символов.

Возвращаемое значение: метод возвращает true, если входная строка содержит только цифры. Если имеется хотя бы один символ, который не является цифрой, то метод должен вернуть false. В случае пустой строки метод также должен вернуть false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №4

1. Создать библиотеку классов, в которой реализован метод, выполняющий проверку на валидность переданного номера телефона. Строка является валидным номером телефона, если строка начинается с « +7», далее в правильном номере телефона следует 10 символов, являющихся цифрами.

Сигнатура метода

```
public static bool IsPhoneNumber(string textString)
```

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать любые символы.

Возвращаемое значение: метод возвращает true, если переданная строка соответствует условиям выше и является корректным номером телефона. В противном случае возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №5

1. Создать библиотеку классов, в которой реализован метод `Censore`, который бы заменял во входной строке все неприличные слова на звездочки `***`. Список неприличных слов задается. Регистр неприличных слов не важен

Сигнатура метода

```
public static string Censore(string textString, List<string> cens)
```

Входное значение

В качестве параметров передаются:

- проверяемая строка `textString`, которая может быть пустой или содержать некоторый текст;

- Список `cens` – это список неприличных слов. Регистр не-приличных слов не важен

Возвращаемое значение: возвращается отфильтрованный текст, в котором все неприличные слова заменены на звездочки `***`

2. Реализовать unit-тестирование созданной библиотеки классов

3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №6

1. Создать библиотеку классов, в которой реализован метод, который бы определял количество слов `word` в тексте `textString`. Предполагается, что словом является любая последовательность букв, ограниченная двумя или одним пробелом (в случае начала или конца строки). Предполагается, что слова проверяются без учета регистра символов.

Сигнатура метода

```
public static int WordCount(string textString, string word)
```

Входные значения: метод принимает две строки: `textString` и `word`. Строка `textString` содержит исходный текст, строка `word` представляет собой слово, которое нужно искать. Для упрощения задачи гарантируется, что переменная `textString` содержит только буквы и пробелы; Значение переменной `word` всегда состоит из букв и содержит по крайней мере два символа.

Возвращаемое значение: метод возвращает целое число - количество слов `word` в тексте `textString`.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №7

1. Создать библиотеку классов, в которой реализован метод `IsPalindrome`, который бы проверял, является ли переданный текст `textString` палиндромом.

Палиндромом будем считать такой текст, который читается одинаково “слева-направо” и “справа-налево”. При этом, в самом тексте не должны учитываться пробельные символы. Например, палиндромом будет являться текст: “А роза упала на лапу Азора”.

Сигнатура метода

```
public static bool IsPalindrome(string textString)
```

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Текст может включать в себя буквы, цифры и пробелы.

Выходное значение: если переданный текст является палиндромом, то возвращается true. В случае пустой строки следует также вернуть true. В остальных случаях возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №8

1. Создать библиотеку классов, в которой реализован метод `GetLetters`, который бы возвращал список всех букв, использованных в переданном тексте `textString`. Особенности списка, который формируется на выходе метода:

- список не должен содержать повторяющихся элементов;
- буквы в списке должны быть приведены к верхнему регистру;
- список должен быть отсортирован в алфавитном порядке (по возрастанию).

Допускается использовать для сортировки библиотечные функции.

Сигнатура метода

```
public static List<char> GetLetters(string textString)
```

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Текст может включать в себя произвольные символы.

Возвращаемое значение: возвращается список символов List<char>. Список должен со-держать буквы искомого текста и удовлетворять требованиям, указанным выше.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №9

1. Создать библиотеку классов, в которой реализован метод IsPangram, которых бы определял, является ли указанный текст textString панграммой.

Панграмма - короткий текст, использующий все или почти все буквы алфавита, по возможности не повторяя их (Wikipedia).

Будем считать панграммой строку, которая содержит все буквы русского алфавита.

Сигнатура метода

```
public static bool IsPangram(string textString)
```

Входное значение: в качестве параметра передается строка, которая может быть пу-стой или содержать некоторый текст. Текст может включать в се-бя произвольные символы.

Возвращаемое значение: возвращает true, если переданный текст является панграммой. В противном случае возвращается false.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №10

1. Создать библиотеку классов, в которой реализован метод для определения самого длинного слова в тексте. Предполагается, что словом является любая последовательность букв, ограниченная двумя или одним небуквенным символом (один символ в случае начала или конца строки). Если текст содержит несколько слов максимальной длины, то следует вернуть то слово, которое встречается в тексте первым.

Сигнатура метода

```
public static string FindLongestWord(string text)
```

Входное значение

В качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Гарантируется, что текст содержит только буквы, пробелы и знаки пунктуации ,, , ! и ?. Обратите внимание, что символы пунктуации не являются частью слова.

Возвращаемое значение: возвращается строка, содержащее самое длинное слово в тексте, приведенное к нижнему регистру. В случае отсутствия слов, следует вернуть пустую строку.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №11

1. Создать библиотеку классов, в которой реализован метод, который бы возвращал список всех слов, которые встречаются в указанном тексте более одного раза.

Сигнатура метода

```
public static List<string> FindNonUniqueWords(string text)
```

Входное значение: в качестве параметра передается строка, которая может быть пустой или содержать некоторый текст. Гарантируется, что текст содержит только буквы, пробелы, переносы \n и знаки пунктуации ,, ., ! и ?.

Возвращаемое значение: метод возвращает список List<string> всех слов, которые встречаются в тексте два или более раз. Слова следует привести к нижнему регистру, а список следует отсортировать в алфавитном порядке.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №12

1. Создать библиотеку классов, в которой реализован метод FindMinimum, который бы возвращал число, соответствующее минимальному числу в заданном массиве целых чисел

Сигнатура метода

```
public static int FindMinimum(List<int> arrayNumbers)
```

Входное значение: в качестве параметра передается массив целых чисел

Возвращаемое значение: метод возвращает целое число, которое является минимальным числом в заданном массиве.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №13

1. Создать библиотеку классов, в которой реализован метод для работы со строкой. Пусть имеется строка вида "N/M". Реализуйте метод, который бы в случае, когда $N > M$ заменил N на минимальное значение (среди чисел M и N). Например,

"1/10" => "1/10"

"251/100" => "100/100"

Сигнатура метода

```
public string FixWrongResult(string textString)
```

Входное значение: в качестве параметра передается строка. Гарантируется (не нуж-но это проверять, значение “заложено” условием задачи), что строка имеет вид "ЧИСЛО1/ЧИСЛО2". Гарантируется, что оба числовых значения корректны, положительны и умещаются в стандартный тип `int`.

Возвращаемое значение: метод возвращает строку, представляющую “исправленный результат”.

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №14

1. Создать библиотеку классов, в которой реализован метод `PasswordStrengthCheker`, который позволяет определить степень сложности пароля.

если пароль имеет длину больше 7 символов сложность повышается на 1 балл;

- если пароль содержит цифры, то сложность повышается ещё на 1 балл;
- если пароль содержит латинские буквы в нижнем регистре, то сложность повышается ещё на 1 балл;
- если пароль содержит Заглавные латинские буквы, то сложность повышается ещё на 1 балл;
- если пароль содержит спецсимволы, то сложность повышается ещё на 1 балл
- если пароль содержит кириллические символы, то выдается исключение «Кириллические символы запрещены при вводе пароля»

Сигнатура метода

```
public int PasswordStrengthCheker(string textString)
```

Входное значение: в качестве параметра передается строка - пароль

Возвращаемое значение: метод возвращает целое число, соответствующее сложности па-роля

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №15

1. Создать библиотеку классов, в которой реализован метод, который по контрольной цифре определяет корректность введенного номера.

Задана строка – номер из 12 цифр определить корректность но-мера по «контрольной» последней цифре.

Для проверки контрольной цифры необходимо умножить каждую цифру числа на соответствующий номер позиции цифры в этой строке. После этого

получившееся число поделить на 11. Если полученный остаток равен 10, тогда последняя цифра числа должна быть равна 1, в противном случае она равна полученному остатку.

Например: для числа 279146358279 необходимо посчитать сумму $1 * 2 + 2 * 7 + 3 * 9 + 4 * 1 + \dots + 12 * 9$. Поделить полученное число на 11. Если полученный остаток равен 10, тогда последняя цифра числа должна быть равна 1, в противном случае она равна полученному остатку.

Сигнатура метода

```
public bool CheckNumber (string textString)
```

Входное значение: в качестве параметра передается строка

Возвращаемое значение: метод возвращает true, если контрольная цифра задана верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №16

1. Создать библиотеку классов, в которой реализован метод, который по контрольной цифре определяет корректность введенного номера.

Число задается в виде строки из цифр. Необходимо получить контрольную цифру для этого числа. Подсчитываются суммы цифр, стоящих на четных и нечетных местах. Сумма цифр, стоящих на четных местах (, суммируется с утроенной суммой цифр, стоящих на нечетных местах.

Если цифра единиц полученного результата равна нулю, то контрольная цифра — 0. Если последняя цифра результата не нуль, то контрольная цифра равна дополнению этой цифры до 10.

Например:

Задана строка из цифр 345678905675

Цифры, стоящие на нечетных местах (справа-налево): 4, 6, 8, 0, 6, 5. Сумма этих цифр равна 29.

Цифры, стоящие на четных местах (справа-налево): 3, 5, 7, 9, 5, 7. Сумма этих цифр равна 36.

$29 * 3 + 36 = 123$. Искомая контрольная цифра 7, т.к. $7 + 3 = 10$.

Сигнатура метода

```
public int CheckDigit(string textString)
```

Входное значение: в качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает целое число, которое является контрольной цифрой для заданной строки, высчитанной по описанному алгоритму

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №17

1. Создать библиотеку классов, в которой реализован метод, выполняющий проверку корректности СНИЛС

Страховой номер индивидуального лицевого счета страхового свидетельства обязательного пенсионного страхования(он же СНИЛС) проверяется на валидность контрольным числом.

СНИЛС имеет вид: «XXX-XXX-XXX YY», где XXX-XXX-XXX — собственно номер, а YY — контрольное число. Алгоритм формирования контрольного числа СНИЛС таков:

□ Проверка контрольного числа Страхового номера проводится только для номеров больше номера 001-001-998

□ Контрольное число СНИЛС рассчитывается следующим образом:

1) Каждая цифра СНИЛС умножается на номер своей позиции (позиции отсчитываются с конца)

2) Полученные произведения суммируются

3) Если сумма меньше 100, то контрольное число равно самой сумме

4) Если сумма равна 100 или 101, то контрольное число равно 00

5) Если сумма больше 101, то сумма делится по остатку на 101 и контрольное число определяется остатком от деления аналогично пунктам 2.3 и 2.4

Например: Указан СНИЛС 112-233-445 95

Проверяем правильность контрольного числа:

цифры номера 1 1 2 2 3 3 4 4 5

номер позиции 9 8 7 6 5 4 3 2 1

Сумма = $1 \times 9 + 1 \times 8 + 2 \times 7 + 2 \times 6 + 3 \times 5 + 3 \times 4 + 4 \times 3 + 4 \times 2 + 5 \times 1 = 95$

$95 \div 101 = 0$, остаток 95. Контрольное число 95 — указано верно

Сигнатура метода

```
public bool CheckMedicalCode(string textString)
```

Входное значение: в качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает true, если СНИЛС задан верно, false – в случае неверного СНИЛС

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №18

1. Создать библиотеку классов, в которой реализован метод, выполняющий расчет контрольной цифры IMEI

IMEI присваивается телефону во время изготовления на заводе. Он служит для идентификации устройства в сети и хранится в прошивке аппарата. Как правило, 15-я цифра IMEI является так называемым контрольным числом, которое позволяет проверить правильность всех остальных цифр.

Вычисление контрольной цифры IMEI

Например IMEI=354190023896443.

1. Сложить все цифры в нечетных позициях;
2. Заменить цифры на четных местах на :
 - o 0=0
 - o 1=2
 - o 2=4
 - o 3=6

o $4=8$

o $5=1$

o $6=3$

o $7=5$

o $8=7$

o $9=9$

3. К полученному числу прибавить результат, полученный в п.1;

4. Если полученное число кратно 10, тогда контрольное число IMEI равно 0. В противном случае контрольная сумма равна числу, которое нужно прибавить к результату, чтобы получить ближайшее большее «круглое» число.

Пример

IMEI=35-419002-389644-3.

Сложить все цифры в нечетных позициях:

$$3+4+9+0+3+9+4 = 32$$

Заменить цифры на четных местах:

$$5,1,0,2,8,6,4 \Rightarrow 1,2,0,4,7,3,8$$

и сложить их:

$$1+2+0+4+7+3+8 = 25$$

К полученному числу прибавить результат п.1.

$$25 + 32 = 57$$

Контрольная сумма равна числу, которое нужно прибавить к результату, чтобы получить ближайшее большее «круглое» число: $60 - 57 = 3$. Итого контрольное число равно 3.

Сигнатура метода

```
public int CheckDigitIMEI(string textString)
```

Входное значение

В качестве параметра передается строка из цифр

Возвращаемое значение: метод возвращает true, если контрольное цифра задана верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

ЗАДАНИЕ №19

1. Создать библиотеку классов, в которой реализован метод, который позволяет определить корректно ли введен код EAN-13 товара

Для этого необходимо посчитать контрольную сумму, которая вычисляется по следующему алгоритму:

1) каждая из цифр номера умножается на соответствующие коэффициенты из таблицы.

k13	k12	k11	k10	k9	k8	k7	k6	k5	k4	k3	k2	k1
1	3	1	3	1	3	1	3	1	3	1	3	1

4600051000057 (сигареты «Прима») — код EAN-13.

$$4 \times 1 + 6 \times 3 + 0 \times 1 + 0 \times 3 + 0 \times 1 + 5 \times 3 + 1 \times 1 + 0 \times 3 + 0 \times 1 + 0 \times 3 + 0 \times 1 + 5 \times 3 + 7 \times 1 =$$

$$4 + 18 + 0 + 0 + 0 + 15 + 1 + 0 + 0 + 0 + 0 + 15 + 7 = 60.$$

2) Если полученная сумма дает остаток от деления на 10 равный нулю, то номер признаётся правильным.

3) $60 \% 10 = 0$ — номер правильный.

Сигнатура метода

```
public bool CheckNumber(string textString)
```

Входное значение: в качестве параметра передается строка из цифр (код)

Возвращаемое значение

Метод возвращает true, если код задан верно, false – в случае неверной контрольной цифры

2. Реализовать unit-тестирование созданной библиотеки классов
3. Создать и задокументировать разработанные тест-кейсы

Фонд оценочных средств
для проведения аттестации

по вариативному модулю СПО

профессиональному модулю среднего профессионального образования: «Разработка
и тестирование информационных систем»

г.Екатеринбург, 2023

Комплект оценочных средств

УСЛОВИЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1. Место выполнения задания: мастерская «Программные решения для бизнеса»
2. Максимальное время выполнения задания: 6 академических часа

ЗАДАНИЕ

Необходимо реализовать приложение для магической лавки «хитрая магическая материя». Приложение позволит упростить ведение магического бизнеса.

При запуске приложения пользователя-мага встречает окно входа. Окно входа изначально должно содержать два поля: имя пользователя и пароль. В случае, если пользовательские данные введены верно, пользователь должен перейти на соответствующее его роли окно (маг первого уровня, маг второго уровня, архимаг).

Если данные были введены неправильно, то окно входа должно быть заколдовано: в специальном поле должно появиться заклинание (captcha), которое нужно написать вместе с именем пользователя и паролем. В случае, если заклинание введено неправильно, вход в систему выполнен не будет, и нужно будет ввести новое заклинание.

Заклинанием может быть не всякий набор символов и не всякое изображение. Оно должно быть показано в строгом соответствии с магическими стандартами:

- во-первых, оно должно включать в себя только слова “абра”, “кадабра”, “сим”, “салабим”, “фокус”, “покус”, “вжух”, “ууу”;
- во-вторых, оно должно состоять из двух таких слов, при этом слова могут повторяться;
- в-третьих, дабы обмануть хитрых гремлинов и големов, слова должны быть либо зачеркнуты, либо наложены друг на друга, либо как-то еще искажены (зашумлены).

Рекомендуется:

1. создать интерфейс приложения

2. сформировать библиотеку классов с методом **GenerateCaptcha()** для генерации волшебного заклинания
3. создать тестовые наборы (чек-лист), содержащий позитивные тесты, негативные тесты из разных классов эквивалентности, достаточные для проверки заклинания (в соответствии с магическими стандартами)