

Документ подписан простой электронной подписью.
Информация о владельце:
ФИО: Костина Лариса Николаевна
Должность: проректор
Дата подписания: 30.12.2024 09:18:20
Уникальный программный ключ:
1800f7d89cf4ea7507265ba593fe87537eb15a6c

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ"

Факультет

Государственной службы и управления

Кафедра

Информационных технологий

"УТВЕРЖДАЮ"

Проректор

 Л.Н. Костина

27.04.2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.15

"Разработка информационных систем"

Направление подготовки 09.03.03 Прикладная информатика

Профиль "Прикладная информатика в управлении корпоративными информационными системами"

Квалификация

БАКАЛАВР

Форма обучения

очная

Общая трудоемкость

6 ЗЕТ

Год начала подготовки по учебному плану

2023

Донецк
2023

Составитель:
ст.препод.



Э.М. Лебезова

Рецензент(ы):
канд. экон. наук, доцент



И.В. Стешенко

Рабочая программа дисциплины "Разработка информационных систем" разработана в соответствии с:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Рабочая программа дисциплины (модуля) составлена на основании учебного плана Направление подготовки 09.03.03 Прикладная информатика Профиль "Прикладная информатика в управлении корпоративными информационными системами", утвержденного Ученым советом ФГБОУ ВО "ДОНАУИГС" от 27.04.2023 протокол № 12.

Срок действия программы: 2023-2027

Рабочая программа рассмотрена и одобрена на заседании кафедры Информационных технологий

Протокол от 20.04.2023 № 9

Заведующий кафедрой:
канд.физ.-мат.наук, доцент Брадул Н.В.



(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2024 - 2025 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2024 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2025 - 2026 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2025 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2026 - 2027 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2026 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент Брадул Н.В.

(подпись)

Визирование РПД для исполнения в очередном учебном году**"УТВЕРЖДАЮ"**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2027 - 2028 учебном году на заседании кафедры Информационных технологий

Протокол от " ____ " _____ 2027 г. №__

Зав. кафедрой канд.физ.-мат.наук, доцент Брадул Н.В.

(подпись)

РАЗДЕЛ 1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЙ

1.1. ЦЕЛИ ДИСЦИПЛИНЫ	
Цель изучения дисциплины – формирование у бакалавров компетенций в области разработки информационных систем.	
1.2. УЧЕБНЫЕ ЗАДАЧИ ДИСЦИПЛИНЫ	
Задачи учебной дисциплины:	
1. Подготовка выпускников к проектно-конструкторской деятельности в области создания и внедрения аппаратных и программных средств объектов профессиональной деятельности в соответствии с техническим заданием и с использованием средств автоматизации проектирования.	
2. Подготовка выпускников к проектно-технологической деятельности в области создания компонентов программных комплексов и баз данных, автоматизации технологических процессов с использованием современных инструментальных средств и технологий программирования.	
3. Подготовка выпускников к комплексным инженерным исследованиям для решения задач, связанных с разработкой аппаратных и программных средств объектов профессиональной деятельности.	
1.3. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОПОП ВО:	Б1.В
<i>1.3.1. Дисциплина "Разработка информационных систем" опирается на следующие элементы ОПОП ВО:</i>	
Проектный практикум	
Базы данных	
Информатика и программирование	
Проектирование информационных систем	
<i>1.3.2. Дисциплина "Разработка информационных систем" выступает опорой для следующих элементов:</i>	
Преддипломная практика	
Разработка программных приложений	
Разработка электронного портала	
1.4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ:	
<i>ПК-5.1: Применяет современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях</i>	
Знать:	
Уровень 1	Знает принципы построения ИС на основе парадигмы ООП
Уровень 2	Знает паттерны проектирования ИС
Уровень 3	Знает современные архитектурные решения для разработки и внедрения информационных систем
Уметь:	
Уровень 1	Умеет создавать программы на языках высокого уровня с помощью объектно-ориентированной и компонентно-ориентированной парадигм
Уровень 2	Умеет создавать ИС с помощью специализированных библиотек и фреймворков
Уровень 3	Умеет создавать ИС с помощью AI и специализированных нейронных систем
Владеть:	
Уровень 1	Владеет архитектурными паттернами для создания информационных систем различных предметных областей
Уровень 2	Владеет программными средами для создания информационных систем различных предметных областей
Уровень 3	Владеет современными облачными и контейнерными технологиями для создания информационных систем различных предметных областей
1.4. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ:	
<i>УК-2.3: Планирует реализацию задач в зоне своей ответственности с учетом имеющихся ресурсов и ограничений, действующих правовых норм</i>	
Знать:	
Уровень 1	Знает методики и технологии разработки информационных систем

Уровень 2	Знает нормативные правовые документы в сфере создания информационного общества
Уровень 3	Знает методики и технологии разработки информационных систем, нормативные правовые документы в сфере создания информационного общества
Уметь:	
Уровень 1	Умеет разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования
Уровень 2	Умеет разрабатывать интерфейсы «человек – электронно-вычислительная машина»
Уровень 3	Умеет разрабатывать проекты информационных систем различного назначения, использовать нормативные и правовые документы в сфере создания информационного общества
Владеть:	
Уровень 1	Владеет навыками сбора информации для инициации проекта в соответствии с полученным заданием
Уровень 2	Владеет навыками анализа и синтеза информации для выполнения проекта в соответствии с полученным заданием
Уровень 3	Владеет навыками разрабатывать, внедрять и адаптировать прикладное программное обеспечение

В результате освоения дисциплины "Разработка информационных систем" обучающийся

3.1	Знать:
	современные технологии и архитектурные решения для разработки и внедрения информационных систем
3.2	Уметь:
	создавать программы на языках высокого уровня с помощью объектно-ориентированной и компонентно-ориентированной парадигм, используя специализированные библиотеки и фреймворки
3.3	Владеть:
	программирования приложений и создания программных прототипов решения прикладных задач

1.5. ФОРМЫ КОНТРОЛЯ

Текущий контроль успеваемости позволяет оценить уровень сформированности элементов компетенций (знаний, умений и приобретенных навыков), компетенций с последующим объединением оценок и проводится в форме: устного опроса на лекционных и семинарских/практических занятиях (фронтальный, индивидуальный, комплексный), письменной проверки (тестовые задания, контроль знаний по разделу, ситуационных заданий и т.п.), оценки активности работы обучающегося на занятии, включая задания для самостоятельной работы.

Промежуточная аттестация

Результаты текущего контроля и промежуточной аттестации формируют рейтинговую оценку работы студента. Распределение баллов при формировании рейтинговой оценки работы студента осуществляется в соответствии с действующим локальным нормативным актом. По дисциплине "Разработка информационных систем" видом промежуточной аттестации является Экзамен

РАЗДЕЛ 2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

2.1. ТРУДОЕМКОСТЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Общая трудоёмкость дисциплины "Разработка информационных систем" составляет 6 зачётные единицы, 216 часов.

Количество часов, выделяемых на контактную работу с преподавателем и самостоятельную работу обучающегося, определяется учебным планом.

2.2. СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
Раздел 1. Методологии и технологии разработки ИС						
Тема № 1. 1 Введение в разработку ИС.	7	4	ПК-5.1	Л1.1Л2.1	0	

Классические, мобильные и веб-приложения /Лек/			УК-2.3	Л2.2Л3.1 Л3.2 Л3.3 Э1		
Тема № 1. 1 Введение в веб разработку. Необходимое ПО, HTML верстка, каскадные таблицы стилей. Общие сведения о DOM, события, способы работы. /Пр/	7	8	ПК-5.1 УК-2.3	Л1.1Л2.1 Л2.2Л3.1 Л3.2 Л3.3 Э1	0	
Тема № 1. 1 Верстка с использованием Bootstrap. /Ср/	7	6	ПК-5.1 УК-2.3	Л1.1Л2.1 Л2.2Л3.1 Л3.2 Л3.3 Э5	0	
Тема № 1.2 Серверные веб-технологии. Понятие бэкэнда и фронтэнда. Стандарт ECMAScript /Лек/	7	4	ПК-5.1 УК-2.3	Л1.1Л2.1 Л2.2Л3.1 Л3.2 Л3.3 Э1	0	
Тема № 1.2 Серверные веб-технологии. Понятие бэкэнда и фронтэнда. Стандарт ECMAScript /Пр/	7	8	ПК-5.1 УК-2.3	Л1.1Л2.1 Л2.2Л3.1 Л3.2 Л3.3 Э1	0	
Тема № 1.2 Серверные веб-технологии. Понятие бэкэнда и фронтэнда. Стандарт ECMAScript /Ср/	7	6	ПК-5.1 УК-2.3	Л1.1Л2.1 Л2.2Л3.1 Л3.2 Л3.3 Э1	0	
Тема № 1.3 Тема № 1.3 Модель приложения на Node.js и библиотека Express /Лек/	7	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.3 Модель приложения на Node.js и библиотека Express /Пр/	7	8	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.3 Модель приложения на Node.js и библиотека Express /Ср/	7	6	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.4. Стандарт REST API /Лек/	7	2	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.4. Создание Nodejs REST API с помощью Express и PostgesSQL /Пр/	7	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.4. Применение и стандарты API. /Ср/	7	10	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э2 Э3	0	
Тема № 1.1. - Тема № 1.4. /Конс/	7	2	ПК-5.1	Л1.1Л2.1Л3	0	

			УК-2.3	.1 Л3.2 Л3.3		
Раздел 2. Разработка ИС с помощью фреймворков						
Тема № 2.1 Тестирование API. Postman /Лек/	8	2	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э7 Э8	0	
Тема № 2.1 Тестирование API. Postman /Пр/	8	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э7 Э8	0	
Тема № 2.1 Тестирование API. Тестирование серверных веб-приложений /Ср/	8	10	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э7 Э8	0	
Тема № 2.2 Документирование API. Markdown /Лек/	8	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э6	0	
Тема № 2.2 Документирование API. Markdown /Пр/	8	8	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э6	0	
Тема № 2.2 Документирование API. Markdown /Ср/	8	18	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э6	0	
Тема № 2.3 Шаблонизаторы для web-разработки /Лек/	8	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э9	0	
Тема № 2.3 Шаблонизаторы для web-разработки /Пр/	8	8	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э9	0	
Тема № 2.3 Системы управления контентом и шаблонизаторы /Ср/	8	22	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э9	0	
Тема № 2.4 Фронтенд фреймворк Vue.js /Лек/	8	4	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э4	0	
Тема № 2.4 Создание приложения с	8	8	ПК-5.1	Л1.1Л2.1Л3	0	

использованием Vue.js /Пр/			УК-2.3	.1 Л3.2 Л3.3 Э4		
Тема № 2.4 Фронтенд фреймворк Vue.js /Ср/	8	23	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3 Э4	0	
Тема № 2.1. - Тема № 2.4. /Конс/	8	2	ПК-5.1 УК-2.3	Л1.1Л2.1Л3 .1 Л3.2 Л3.3	0	

РАЗДЕЛ 3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В процессе освоения дисциплины используются следующие образовательные технологии: лекции (Л), практические занятия (ПР), самостоятельная работа студентов (СР) по выполнению различных видов заданий.

1. В процессе освоения дисциплины используются следующие интерактивные образовательные технологии: проблемная лекция (ПЛ). Лекционный материал представлен в виде слайд-презентации в формате «Power Point». Для наглядности используются материалы различных научных и технических экспериментов, справочных материалов, научных статей т.д. В ходе лекции предусмотрена обратная связь со студентами, активизирующие вопросы, просмотр и обсуждение видеofilмов. При проведении лекций используется проблемно-ориентированный междисциплинарный подход, предполагающий творческие вопросы и создание дискуссионных ситуаций.

2. При изложении теоретического материала используются такие методы:

- монологический;
- показательный;
- диалогический;
- эвристический;
- исследовательский;
- проблемное изложение.

3. Используются следующие принципы дидактики высшей школы:

- последовательность обучения;
- систематичность обучения;
- доступность обучения;
- принцип научности;
- принципы взаимосвязи теории и практики;
- принцип наглядности и др.

В конце каждой лекции предусмотрено время для ответов на проблемные вопросы.

4. Практические занятия проходят в компьютерных классах для приобретения и закрепления практических навыков для освоения дисциплины в полной мере. Выполняя задания для практических работ, студенты осваивают новые технологии, изучают современные языки программирования и накапливают опыт разработки информационных систем.

5. Самостоятельная работа предназначена для внеаудиторной работы студентов, связанной с конспектированием источников, учебного материала, изучением дополнительной литературы по дисциплине, подготовкой к текущему и семестровому контролю, а также выполнением индивидуального задания в форме реферата, эссе, презентации, эмпирического исследования.

РАЗДЕЛ 4. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

4.1. Рекомендуемая литература

1. Основная литература

	Авторы,	Заглавие	Издательство, год
Л1.1	Т. В. Гвоздева	Проектирование информационных систем. : Лабораторный практикум : учебное пособие (116 с.)	Санкт-Петербург : Лань, 2022

2. Дополнительная литература

	Авторы,	Заглавие	Издательство, год
Л2.1	Э. Лок ; перевод с английского Д. А. Беликова	ASP.NET Core в действии: руководство (906 с.)	Москва : ДМК Пресс, Лань : электронно-библиотечная система., 2021
Л2.2	И.Б. Государев	Введение в веб-разработку на языке JavaScript [Электронный ресурс]: учебное пособие (144 с.)	Санкт-Петербург : Лань, 2019

3. Методические разработки

	Авторы,	Заглавие	Издательство, год
Л3.1	Лебезова Э.М.	Методические рекомендации для проведения практических занятий по учебной дисциплине «Разработка информационных систем» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (35 с.)	Донецк : ДОНАУИГС, 2022
Л3.2	Лебезова Э.М.	Методические рекомендации по организации самостоятельной работы обучающихся по учебной дисциплине «Разработка информационных систем» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (25 с.)	Донецк : ДОНАУИГС, 2022
Л3.3	Лебезова Э.М.	Конспект лекций по учебной дисциплине «Разработка информационных систем» (для студентов образовательного уровня «бакалавр» направления подготовки 09.03.03 Прикладная информатика) / сост. Лебезова Э.М. : учебное пособие (150 с.)	Донецк : ДОНАУИГС, 2022

4.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Современный открытый учебник Javascript.ru	javascript.ru
Э2	Вводное руководство по Express	https://huncode.com/js/express
Э3	Документация по Nodejs	https://nodejs.org/en/
Э4	Документация по Vuejs	https://cli.vuejs.org/
Э5	Документация по Bootstrap	https://getbootstrap.com/
Э6	Справочные материалы по Markdown	https://learn.microsoft.com/ru-ru/contribute/markdown-reference
Э7	Стратегия тестирования REST API	https://habr.com/ru/articles/568360/
Э8	Документация по Postman	https://gb.ru/posts/kak-testirovat-api-ili-postman-dlya-chajnikov
Э9	Шаблонизаторы для фронтенд-разработки	https://proglib.io/p/top-10-shablonizatorov-dlya-frontend-razrabotki-2023-01-17

4.3. Перечень программного обеспечения

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства:

При изучении дисциплины используется ПО в составе пакета OS MS Windows, MS Office, среда разработки Visual Studio (не ниже 15 версии), среда разработки Visual Code, сервера БД – MySQL и MSSql.

Операционная система - Microsoft Windows

Пакет настольных приложений Microsoft Office - (Word, Excel, PowerPoint, OneNote, Outlook, Publisher, Access)

Средства для разработки и проектирования - Android Studio, Microsoft Visual Code, FotoScape, Figma, Avocode

Онлайн системы - GitHub, Markdown

СУБД - MySQL, PostgesSQL

4.4. Профессиональные базы данных и информационные справочные системы

В процессе изучения дисциплины используются возможности информационно-справочной системы портала <http://window.edu.ru/>.

Сайт с учебными курсами по различным аспектам разработки информационных систем. <http://www.intuit.ru>

4.5. Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения занятий лекционного, семинарского типа, групповых занятий и консультаций, текущего контроля и промежуточной аттестации: аудитория № 704 учебный корпус № 1.

- компьютеры (16) с минимальными характеристиками:

Процессор - Intel Core2Duo 2.4GHz

ОЗУ - 8GB

Жёсткий диск - 7200 RPM

Сетевые возможности - доступ к локальной сети 100 Mb/s, доступ к интернет.;

- программное обеспечение - Microsoft Office 2010 (лицензия № 47556582 от 19.10.2010г., лицензия № 49048130 от 19.09.2011);

- комплект мультимедийного оборудования: ноутбук, мультимедийный проектор, экран;

- программное обеспечение - Windows 8.1 Professional x86/64 (академическая подписка DreamSpark Premium), LibreOffice 4.3.2.2 (лицензия GNU LGPL v3+ и MPL2.0);

- специализированная мебель: рабочее место преподавателя, рабочие места обучающихся (32), стационарная доска.

2. Помещения для самостоятельной работы с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно образовательную среду организации:

читальные залы, учебные корпуса 1, 6. Адрес: г. Донецк, ул. Челюскинцев 163а, г. Донецк, ул. Артема 94.

Компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду (ЭИОС ГОУ ВПО ДОНАУИГС) и электроннобиблиотечную систему (ЭБС IPRbooks), а также возможностью индивидуального неограниченного доступа

обучающихся в ЭБС и ЭИОС посредством Wi-Fi с персональных мобильных устройств.

Сервер: AMD FX 8320/32Gb(4x8Gb)/4Tb(2x2Tb). На сервере установлена свободно распространяемая операционная система DEBIAN 10. MS Windows 8.1 (Лицензионная версия операционной системы подтверждена сертификатами подлинности системы Windows на корпусе ПК), MS Windows XP (Лицензионная версия операционной системы подтверждена сертификатами подлинности системы Windows на корпусе ПК), MS Windows 7 (Лицензионная версия операционной системы подтверждена сертификатами подлинности системы Windows на корпусе ПК), MS Office 2007 Russian OLP NL AE (лицензии Microsoft № 42638778, № 44250460), MS Office 2010 Russian (лицензии Microsoft № 47556582, № 49048130), MS Office 2013 Russian (лицензии Microsoft № 61536955, № 62509303, № 61787009, № 63397364), Grub loader for ALT Linux (лицензия GNU LGPL v3), Mozilla Firefox (лицензия MPL2.0), Moodle (Modular Object-Oriented Dynamic Learning Environment, лицензия GNU GPL), IncScape (лицензия GPL 3.0+), PhotoScape (лицензия GNU GPL), 1С ERP УП, 1С ЗУП (бесплатные облачные решения для образовательных учреждений от 1Cfresh.com), OnlyOffice 10.0.1 (SaaS, GNU Affero General Public License3)

РАЗДЕЛ 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

Вопросы к экзамену

1. Охарактеризуйте простой метод водопада и задачи каждого этапа
2. Понятие и принципы Agile-подхода
3. Принципы Extreme Programming
4. Охарактеризуйте методологию SCRUM
5. Принципы и процессы RUP
6. Способы выявления классов на этапе предварительного анализа системы для создания модели предметной области.
7. Понятие и источники ограничений проекта информационной системы.
8. Понятие требований к системе, способы выявления требований. Достоинства и недостатки различных методов сбора данных о системе.
9. Понятие словаря данных, виды его описаний и содержимое словаря данных. Понятие композиции данных и атрибуты словаря и потока данных.
10. Понятие и символы диаграмм потока данных, формирование перечня транзакций и таблица событий. Приведите пример
11. Понятие элементарного процесса, виды логических элементарных процессов и критерии элементарного уровня.

12. Понятие диаграммы вариантов использования, элементы и правила разработки диаграммы вариантов использования. Поток событий и его описание.
13. Понятие стереотипов классов; дайте характеристику стереотипов.
14. Понятие и разделы технико-экономического обоснования (ТЭО) ИС. Характеристика рассматриваемых вопросов технического аспекта ТЭО.
15. Понятие и разделы технико-экономического обоснования (ТЭО) ИС. Материальные и нематериальные выгоды. Составляющие выгод применения ИС.
16. Структура видов деятельности менеджера проекта в соответствии с РМВОК.
17. Понятие рисков и управление рисками. Классификация рисков. Процесс управления рисками, приведите и прокомментируйте схему. Приведите примеры индикаторов рисков.
18. Потребности проекта сети для ИС. Понятие и назначение компьютерной сети. Понятие LAN, WAN, Router, Internet, Intranet и Extranet. Приведите пример возможной конфигурации сети для двух распределенных мест эксплуатации ИС.
19. Приведите перечень широко используемых подходов (pattern) для архитектуры приложения, дайте характеристику функциям приложения ИС. Централизованные серверные архитектуры на основе одного компьютера.
20. Понятие кластерной и мультимедийной архитектуры. Разновидности кластеров.
21. Понятие распределенной клиент-серверной архитектуры. Двух-уровневая клиент-серверная архитектура, преимущества и недостатки. Понятие толстый клиент/толстый сервер.
22. Трех-уровневая архитектура приложения, распределение функций между уровнями. Характеристика N-уровневой модели архитектуры. Приведите пример N-уровневой архитектуры.
23. Принципы проектирования пользовательского интерфейса, характеристика хорошего интерфейса и рекомендации по его созданию.
24. Реляционная модель данных (РМД). Структуризация данных в РМД. Свойства отношения.
25. Основные операции в РМД. Ограничения целостности. Достоинства и недостатки РМД.
26. Проектирование баз данных. Требования к проекту базы данных. Этапы проектирования.
27. Методы инфологического проектирования. Метод "сущность-связь".
28. Нормализация отношений (до 4НФ).
29. Распределенные базы данных (РБД): принципы построения, достоинства и недостатки.
30. Методы поддержки распределенных баз данных: распределенные ограничения целостности; распределенные запросы.
31. Методы поддержки распределенных баз данных: фрагментация.
32. Методы поддержки распределенных баз данных: репликация.
33. Методы поддержки распределенных баз данных: распределенные транзакции; механизм двухфазной фиксации.

5.2. Темы письменных работ

Письменные работы не предусмотрены

5.3. Фонд оценочных средств

Фонд оценочных средств дисциплины "Разработка информационных систем" разработан в соответствии с локальным нормативным актом ФГБОУ ВО "ДОНАУИГС".

Фонд оценочных средств дисциплины "Разработка информационных систем" в полном объеме представлен в виде приложения к данному РПД.

5.4. Перечень видов оценочных средств

Тестирование (проводится на практических занятиях; контроль знаний раздела учебной дисциплины)

Практические задания (выполняются на практических занятиях за компьютером)

Индивидуальные задания (разноуровневые задачи и задания)

РАЗДЕЛ 6. СРЕДСТВА АДАПТАЦИИ ПРЕПОДАВАНИЯ ДИСЦИПЛИНЫ К ПОТРЕБНОСТЯМ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей:

1) с применением электронного обучения и дистанционных технологий.

2) с применением специального оборудования (техники) и программного обеспечения, имеющихся в ФГБОУ ВО "ДОНАУИГС".

В процессе обучения при необходимости для лиц с нарушениями зрения, слуха и опорно-двигательного аппарата предоставляются следующие условия:

- для лиц с нарушениями зрения: учебно-методические материалы в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в

аудиоформат); индивидуальные задания и консультации.

- для лиц с нарушениями слуха: учебно-методические материалы в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: учебно-методические материалы в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

РАЗДЕЛ 7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО УСВОЕНИЮ ДИСЦИПЛИНЫ

К числу важнейших умений, являющихся неотъемлемой частью успешного учебного процесса, относится умение работать с различными литературными источниками, содержание которых так или иначе связано с изучаемой дисциплиной.

Подготовку к любой теме курса рекомендуется начинать с изучения презентационных материалов или учебной литературы, в которых дается систематизированное изложение материала, разъясняется смысл разных терминов и сообщается об изменениях в подходах к изучению тех или иных проблем данного курса.

В курсе широко используются актуальные и эффективные техники для более качественного обучения, социализации:

- обсуждения тем в специальных группах популярных социальных сетей;
- командная работа;
- удаленные технические консультации и видеоконференции;
- наглядные демонстрации современных интернет технологий.

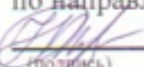
Методические указания по организации самостоятельной работы


Самостоятельная работа по дисциплине организована в следующих видах:

1. изучение теоретического материала по заданной теме;
2. анализ методов решения поставленной задачи;
3. выполнение индивидуальных заданий;
4. оценка достоверности полученных результатов;
5. отчет перед преподавателем по теоретической и практической части индивидуальной работы;
6. самостоятельная работа над индивидуальными заданиями представлены в виде элементов электронного курса в системе elearn

<http://elearn.dsum.org/course/view.php?id=127>

ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ГЛАВЕ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ»

СОГЛАСОВАНО
Руководитель ОПОП ВО
по направлению подготовки
 И.Л. Семичастный
(подпись) (инициалы, фамилия)
29 августа 2022г.

УТВЕРЖДАЮ
Заведующий кафедрой
информационных технологий
 Н.В. Брадул
(подпись) (инициалы, фамилия)
29 августа 2022г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

«Программирование»

Направление подготовки	09.03.03 Прикладная информатика
Профиль	«Прикладная информатика в управлении корпоративными информационными системами»
Квалификация	бакалавр
Форма обучения	очная
Год начала подготовки по учебному плану	2022
Составитель	старший преподаватель, Лебезова Э.М.

Рассмотрено
на заседании ПМК кафедры
«Прикладная информатика и
информационные технологии»
Протокол № 1 от 29.08.2022 г.

Одобрено на заседании кафедры
и рекомендовано к утверждению
Протокол № 1 от 29.08.2022г.

Донецк
2022

РАЗДЕЛ 1.
ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине «Программирование»

1.1. Основные сведения об учебной дисциплине

Таблица 1

Характеристика учебной дисциплины
(сведения соответствуют разделу РПУД)

Образовательная программа	бакалавриат
Направление подготовки	09.03.03 Прикладная информатика «Прикладная информатика в управлении корпоративными информационными системами»
Количество разделов учебной дисциплины	2
Часть образовательной программы	Б1.О.11
Формы текущего контроля	Индивидуальные задания, практические задания, контрольные задания, тестирование.
<i>Показатели</i>	Очная форма обучения
Количество зачетных единиц (кредитов)	6
Семестр	4
Общая трудоемкость (академ. часов)	216
Аудиторная контактная работа:	66
Лекционные занятия	32
Практические занятия	32
Консультации	2
Самостоятельная работа	150
<i>Форма промежуточной аттестации</i>	зачёт с оценкой

1.2. Перечень компетенций с указанием этапов формирования в процессе освоения образовательной программы.

Таблица 2

Перечень компетенций и их элементов

Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
ПК-2.1	Разрабатывает и адаптирует прикладное программное обеспечение	<i>Знать:</i>	
		1. Знать современные информационные технологии и программные средства для разработки прикладного программного обеспечения	ПК 2.1 З-1
		2. Знать методики адаптации программного обеспечения	ПК 2.1 З-2
		3. Знать паттерны проектирования программного обеспечения	ПК 2.1 З-3
		<i>Уметь:</i>	
		1. Уметь выбирать современные информационные технологии и программные средства для разработки прикладного программного обеспечения	ПК 2.1 У-1
		2. Уметь применять методики адаптации программного обеспечения с использованием	ПК 2.1 У-2

Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
		платформы .NET и языка программирования C#	
		3. Уметь разрабатывать и адаптировать прикладное программное обеспечение на основе паттернов проектирования	ПК 2.1 У-3
		Владеть:	
		1. Владеть навыками применения современных информационных технологий и программных средств при разработке и адаптации прикладного программного обеспечения	ПК 2.1 В-1
		2. Владеть навыками разработки программного обеспечения различного назначения, в том числе кроссплатформенного	ПК 2.1 В-2
		3. Владеть технологиями .NET и Core .NET, языком программирования C#	ПК 2.1 В-3
ОПК-7.1	Разрабатывает алгоритмы и программы, пригодные для практического применения	Знать:	
		1. Знать базовые и продвинутое алгоритмы для решения практических задач	ОПК 7.1 3-1
		2. Знать базовые структуры данных для реализации алгоритмов и	ОПК 7.1 3-2

Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
		программ, пригодных для практического применения	
		3. Знать технологии реализации базовых алгоритмов программирования и структур данных на языке программирования C#	ОПК 7.1 З-3
		Уметь:	
		1. Уметь создавать программы на языках высокого уровня с помощью платформы .NET	ОПК 7.1 У-1
		2. Уметь создавать программы с помощью объектно-ориентированной парадигмы	ОПК 7.1 У-2
		3. Уметь анализировать и реализовывать базовые алгоритмы программирования и структуры данных	ОПК 7.1 У-3
		Владеть:	
		1. Владеть интегрированными средами разработки для создания прикладных программ	ОПК 7.1 В-1
		2. Владеть архитектурными паттернами для создания программ, пригодных для	ОПК 7.1 В-2

Код индикатора достижения компетенции	Формулировка индикатора достижения компетенции	Элементы компетенции	Индекс элемента
		практического применения	
		3. Владеть навыками разработки алгоритмов для проведения экспериментальных исследований в области информатики	ОПК 7.1 В-3

Таблица 3

Этапы формирования компетенций в процессе освоения основной образовательной программы

№ п/п	Контролируемые разделы (темы) учебной дисциплины	Этапы формирования компетенций, номер семестра	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
Раздел №1. Технологии создания приложений на платформе .NET				
1.	Тема 1.1. Обзор технологий платформы .NET	1	ПК-2 З-1 ОПК-7 У-1 ОПК-7 В-1	Индивидуальные задания, практические задания, тест
2.	Тема 1.2. Обработка событий и исключительных ситуаций в C#. Технология LINQ	1	ПК-2 З-1 ПК-2 У-1 ПК-2 В-1 ОПК-7 З-1 ОПК-7 З-4	Индивидуальные задания, практические задания, тест
3	Тема 1.3. Многопоточность. Поток, асинхронные	1	ПК-2 У-1 ПК-2 В-1 ОПК-7 З-1 ОПК-7 У-2	Индивидуальные задания, практические задания, тест

	вычисления. Создание многопоточных приложений			
Раздел № 2. Создание программных продуктов различного назначения				
4.	Тема 2.1. Подходы создания GUI в MS VisualStudio	1	ПК-2 У-1 ПК-2 В-1 ОПК-7 З-1 ОПК-7 В-1	Индивидуаль- ные задания, практические задания, тест
5.	Тема 2.2. Графические возможности С# и платформы .NET	1	ПК-2 З-1 ПК-2 У-1 ОПК-7 У-1 ОПК-7 В-1	Индивидуаль- ные задания, практические задания, тест
6.	Тема 2.3. Разработка программ для работы с базами данных MS SQL и My SQL	1	ПК-2 У-1 ПК-2 В-1 ОПК-7 В-1	Индивидуаль- ные задания, практические задания, тест

1.3. Описание показателей и критериев оценивания компетенций на различных этапах формирования, описание шкалы оценивания.

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
Знает	ОПК-7 З-1, ОПК-7 З-2, ОПК-7 З-3, ОПК-7 З-4, ПК-2 З-1	Отлично	90-100	теоретическое содержание дисциплины освоено полностью, без пробелов; необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные рабочей программой дисциплины задания выполнены, качество их выполнения оценено числом
Умеет	ОПК-7 У-1, ПК-2 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1, ПК-2 В-1			

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
				баллов, близким к максимальному
Знает	ОПК-7 3-1, ОПК-7 3-2, ОПК-7 3-3, ОПК-7 3-4, ПК-2 3-1	Хорошо	75-89	теоретическое содержание дисциплины освоено полностью, без пробелов; некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные рабочей программой дисциплины задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Умеет	ОПК-7 У-1, ПК-2 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1, ПК-2 В-1			
Знает	ОПК-7 3-1, ОПК-7 3-2, ОПК-7 3-3, ОПК-7 3-4, ПК-2 3-1	Удовлетворительно	60-74	теоретическое содержание дисциплины освоено частично, но пробелы не носят существенного характера; необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных рабочей программой дисциплины учебных задания выполнено, некоторые из выполненных
Умеет	ОПК-7 У-1, ПК-2 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1, ПК-2 В-1			

Дескриптор компетенции	Показатель оценки	Шкалы оценивания		Критерии оценивания
		Государственная	Баллы	
1	2	3	4	5
				заданий содержат ошибки
Знает	ОПК-7 З-1, ОПК-7 З-2, ОПК-7 З-3, ОПК-7 З-4, ПК-2 З-1	Неудовлетворительно	0-59	теоретическое содержание дисциплины не освоено полностью; необходимые практические навыки работы не сформированы, все предусмотренные рабочей программой задания выполнены с грубыми ошибками либо совсем не выполнены, качество их выполнения оценено числом баллов, близким к минимальному
Умеет	ОПК-7 У-1, ПК-2 У-1, ОПК-7 У-2			
Владеет	ОПК-7 В-1, ПК-2 В-1			

РАЗДЕЛ 2 Текущий контроль

Текущий контроль знаний используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной работой) обучающихся. В условиях балльно-рейтинговой системы контроля результаты текущего оценивания обучающегося используются как показатель его текущего рейтинга. Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе преподавателя. Данный вид контроля стимулирует у обучающегося стремление к систематической самостоятельной работе по изучению учебной дисциплины.

2.1. Описание оценочных средств по видам заданий текущего контроля

Распределение баллов по рейтинговой системе оценивания
по видам учебной деятельности

Сумма баллов по разделу	Раздел 1		Раздел 2		Сумма баллов за дисциплину 100
Темы	Т.1.1-1.3	Контроль знаний раздела учебной дисциплины - 5	Т. 2.1 - 2.3	Контроль знаний раздела учебной дисциплины - 5	
Виды работ:					
Практические работы	20		20		
Индивидуальные задания	20		20		
Контрольные задания	5	5			
Сумма баллов	45		45		

2.1.1. Рекомендации по оцениванию индивидуальных и практических работ обучающихся

Максимальное количество баллов (государственная оценка)	Критерии
10-9 (отлично)	выставляется обучающемуся: если выполнены все пункты работы самостоятельно, без ошибок, если предложен более рациональный алгоритм решения задачи.
8-7 (хорошо)	выставляется обучающемуся: если самостоятельно выполнены все пункты работы, допущены незначительные ошибки, если предложен более рациональный алгоритм решения задачи.
6-5 (удовлетворительно)	выставляется обучающемуся: если самостоятельно (или с помощью преподавателя) выполнены все пункты работы, допущены грубые ошибки.
4 и менее (неудовлетворительно)	выставляется обучающемуся: если с помощью преподавателя выполнены не все пункты работы, допущены грубые ошибки.

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

Индивидуальная работа №1

Раздел №1. Технологии создания приложений на платформе .NET.

Тема 1.1. -1.3

Вариант 1:

Даны Таблицы Factories (заводы), Units (установки), Tanks (резервуары)

Id	Name	Description			
1	НПЗ№1	Первый нефтеперерабатывающий завод			
2	НПЗ№2	Второй нефтеперерабатывающий завод			

Id	Name	Description	FactoryId
1	ГФУ-2	Газофракционирующая установка	1
2	АВТ-6	Атмосферно-вакуумная трубчатка	1
3	АВТ-10	Атмосферно-вакуумная трубчатка	2

Id	Name	Description	Volume	MaxVolume	UnitId
1	Резервуар 1	Надземный - вертикальный	1500	2000	1
2	Резервуар 2	Надземный - горизонтальный	2500	3000	1
3	Дополнительный резервуар 24	Надземный - горизонтальный	3000	3000	2

I d	Name	Description	Volum e	MaxVolum e	UnitI d
4	Резервуар 35	Надземный - вертикальный	3000	3000	2
5	Резервуар 47	Подземный - двустенный	4000	5000	2
6	Резервуар 256	Подводный	500	500	3

1. Создать классы, описывающие структуру каждой из приведённых выше таблиц
2. Создать экземпляры объектов из таблицы. Вы можете создавать объекты с помощью new, прямо в коде определив значения из приложенных таблиц. По желанию, можно реализовать чтение и десериализацию из json-файлов (потребуется продумать структуру и подготовить json файлы с указанными в таблицах данными)
3. Реализовать функции, возвращающие результат согласно комментариям
 - используйте циклы, условные операторы и оператор switch
 - для более высокой оценки, используйте LINQ для более лаконичных запросов при фильтрации, выборе или суммировании значений
4. Осуществить вывод в консоль всех резервуаров, включая имена цеха и фабрики, в которых они числятся
5. Вывести общую сумму загрузки всех резервуаров
6. * Осуществить возможность поиска по наименованию в коллекции, например через ввод в консоли
7. ** Придумать структуру и выгрузить все объекты в json файл
8. *** Считать данные таблиц Excel напрямую, используя любую библиотеку

За основу можно взять код ниже:

Пример кода

```
class Program
{
    static void Main(string[] args)
    {
        var tanks = GetTanks();
        var units = GetUnits();
        var factories = GetFactories();
    }
}
```



```

    Console.WriteLine($"Количество резервуаров: {tanks.Length}, установок:
{units.Length}");

    var foundUnit = FindUnit(units, tanks, "Резервуар 2");
    var factory = FindFactory(factories, foundUnit);

    Console.WriteLine($"Резервуар 2 принадлежит установке {foundUnit.Name} и заводу
{factory.Name}");

    var totalVolume = GetTotalVolume(tanks);
    Console.WriteLine($"Общий объем резервуаров: {totalVolume}");
}

// реализуйте этот метод, чтобы он возвращал массив резервуаров, согласно
приложенным таблицам
// можно использовать создание объектов прямо в C# коде через new, или читать из
файла (на своё усмотрение)
public static Tank[] GetTanks()
{
    // ваш код здесь
}
// реализуйте этот метод, чтобы он возвращал массив установок, согласно приложенным
таблицам
public static Unit[] GetUnits()
{
    // ваш код здесь
}
// реализуйте этот метод, чтобы он возвращал массив заводов, согласно приложенным
таблицам
public static Factory[] GetFactories()
{
    // ваш код здесь
}

// реализуйте этот метод, чтобы он возвращал установку (Unit), которой
// принадлежит резервуар (Tank), найденный в массиве резервуаров по имени
// учтите, что по заданному имени может быть не найден резервуар
public static Unit FindUnit(Unit[] units, Tank[] tanks, string unitName)
{
    // ваш код здесь
}

// реализуйте этот метод, чтобы он возвращал объект завода, соответствующий
установке
public static Factory FindFactory(Factory[] factories, Unit unit)
{
    // ваш код здесь
}

// реализуйте этот метод, чтобы он возвращал суммарный объем резервуаров в массиве
public static int GetTotalVolume(Tank[] units)
{
    // ваш код здесь
}
}

/// <summary>
/// Установка
/// </summary>
public class Unit
{
    //..

```

```

}

/// <summary>
/// Завод
/// </summary>
public class Factory
{
    //..
}

/// <summary>
/// Резервуар
/// </summary>
public class Tank
{
    //..
}

```

Код выполненного задания опубликовать в свой git-репозиторий. Каждое из следующих заданий выполняйте в отдельной ветке этого репозитория, оформляя merge request в основную ветку.

Индивидуальная работа №2

Раздел № 2. Создание программных продуктов различного назначения Тема 2.1-2.3.

В приложении, реализованном в рамках предыдущей задачи, реализовать авторизацию на основе JSON Web Token (JWT).

Должны быть созданы необходимые таблицы в БД с пользователями и их правами (клэймами).

При первоначальной миграции БД должен быть добавлен пользователь (seed data) с логином "admin", пароль "pwd123"

Пароли не должны храниться в БД в открытом виде (хранение в БД открытых паролей - грубейшая ошибка разработчиков приложений).

Токен должен быть подписан секретным ключом, который хранится в appsettings.json

Время действия токена - 2 часа

Предоставлен API для авторизации, получения информации о текущем авторизованном пользователе и смены пароля:

- GET api/user/current - получение информации о текущем пользователе (требует авторизации)
- POST api/user/auth - авторизация пользователя по логину и паролю (доступен для анонимных пользователей)
- {
- "login": "admin",

- "password": "pwd123"
- }

Сгенерированный токен может быть возвращён через заголовок Set-Cookie либо в теле ответа как одно из полей в модели.

POST api/user/password/update - обновить пароль авторизованного пользователя (требуется авторизация)

```
{
  "login": "admin",
  "currentPassword": "pwd123",
  "newPassword": "pwd321"
}
```

Все существующие методы управления и чтения установок и резервуаров (UnitController и TankController) должны требовать авторизацию. Если запрос не содержит токена, либо представленный в HTTP-заголовках запроса Bearer-токен невалидный (истёк, неверно подписан) - должен возвращаться ответ 401 Unauthorized

Управление установками, а также управление резервуарами требует наличия соответствующего типа клэймов у авторизованного пользователя, в противном случае возвращается ответ 403 Forbidden)

Дополнительно реализовать методы управления пользователями - создание и удаление пользователя, управление правами пользователя (для редактирования пользователей, авторизованный пользователь также должен обладать соответствующими правами)

Практическая работа № 1

Раздел №1. Технологии создания приложений на платформе .NET

Тема 2.2

1. Построить три класса (базовый и 2 потомка), описывающих некоторых работников с почасовой оплатой (один из потомков) и фиксированной оплатой (второй потомок). Описать в базовом классе абстрактный метод для расчета среднемесячной заработной платы. Для «повременщиков» формула для расчета такова: «среднемесячная заработная плата = количество отработанных дней * 8 * почасовую ставку», для работников с фиксированной оплатой «среднемесячная заработная плата = фиксированному окладу».

2. Создать список сотрудников, рассчитать их з/п и с помощью технологии LINQ осуществить поиск сотрудника по имени.

3.3. Рекомендации по оцениванию устных ответов обучающихся

Оценка «5» - 5 баллов - ставится, если обучающийся:

- 1) полно и аргументировано отвечает по содержанию вопроса;
- 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике;
- 3) умеет достаточно глубоко и доказательно обосновать свои суждения и применяемый инструментарий для решения задания;

Оценка «4» - 4 балла - ставится, если обучающийся дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 ошибки, которые сам же исправляет.

Оценка «3» - 3 балла - ставится, если обучающийся обнаруживает знание и понимание основных положений данного задания, но:

- 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;
- 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и применяемый инструментарий для решения задания;
- 3) излагает материал непоследовательно и допускает ошибки.

Оценка «2» - 1-2 балла - ставится, если обучающийся обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает применяемый инструментарий для решения задания. Оценка «2» отмечает такие недостатки в подготовке обучающегося, которые являются серьезным препятствием к успешному овладению последующим материалом.

ТИПОВЫЕ ВОПРОСЫ ДЛЯ САМОПОДГОТОВКИ

Контролируемые разделы (темы) учебной дисциплины	Вопросы, выносимые на самостоятельное обучение по разделам дисциплины
Раздел №1. Технологии создания приложений на платформе .NET	
Тема 1.1. Обзор технологий платформы .NET	<ol style="list-style-type: none">1. JSON - что за формат, какие правила? Чем отличается от XML - преимущества и недостатки.2. Что такое Newtonsoft Json (на смену которому пришёл System.Text.Json от MS)?3. Что такое сериализация и десериализация?4. Кодировки, unicode, utf-8, utf-16. Чем UTF-8 отличается от ASCII, насколько он с ней совместим?5. Паттерн MVC. Что означает каждая буква, за что отвечает.

	<ol style="list-style-type: none"> 6. Entity Framework. Что это такое, история развития, версии. 7. Регулярные выражения. 8. HTTP: 9. методы (какие бывают, какая семантика у каждого из них). Чем отличается GET и POST методы? 10. заголовки (что это, зачем и как используется. Часто используемые заголовки: Authorization, Set-Cookie, Content-Type, Access-Control-Allow-Origin) 11. тело запроса (что может содержать) 12. REST API. Основные принципы проектирования API, соответствующего REST. 13. Сетевая модель OSI: 14. Протоколы TCP, UDP. Основные принципы и отличия 15. HTTP/HTTPS - сертификаты, корневые центры. Зачем нужен HTTPS, от чего позволяет защищаться. 16. IPv4, IPv6 17. Веб-сокеты. Как происходит установление соединения. Принципы использования. 18. Unit-тесты. 19. Fluent Assertions.
<p>Тема 1.2. Обработка событий и исключительных ситуаций в C#. Технология LINQ</p>	<ol style="list-style-type: none"> 1. Обработка исключений 2. Делегаты, события, лямбды 3. Коллекции 4. Парсинг JSON и применение LINQ
<p>Тема 1.3. Многопоточность. Поток, асинхронные вычисления. Создание многопоточных приложений</p>	<ol style="list-style-type: none"> 1. Обобщения (Generics), параметры типа 2. IEnumerable и IEnumerable<T> 3. Структуры данных (массив, динамический массив, хэш-таблица, связный список, дерево) 4. IQueryable, EF Core 5. Использование стороннего API. 6. Реализация WebApi на ASP.NET Core, использование websockets 7. API-сервис DaData 8. Отправка HTTP-запросов из приложения на C# 9. Использование Fiddler/Postman

	<ul style="list-style-type: none"> 10. Создание простого WebApi 11. Использование SignalR 12. Использование веб-сокетов напрямую 13. Универсальные шаблоны 14. Асинхронность
Раздел 2. Создание программных продуктов различного назначения	
Тема 2.1. Подходы создания GUI в MS VisualStudio	<ul style="list-style-type: none"> 1. Обобщения (Generics), параметры типа 2. IEnumerable и IEnumerable<T> 3. Структуры данных (массив, динамический массив, хэш-таблица, связный список, дерево) 4. IQueryable, EF Core
Тема 2.2. Графические возможности C# и платформы .NET	<ul style="list-style-type: none"> 1. Перегрузка методов, необязательные параметры со значениями по умолчанию, out и ref аргументы 2. Интерфейс IDisposable, конструкция using 3. Атрибуты 4. Garbage Collector
Тема 2.3. Разработка программ для работы с базами данных MS SQL и My SQL	<ul style="list-style-type: none"> 1. Web API 2. Entity framework core 3. Dependency injection 4. Безопасность веб-приложений 5. Безопасное хранение паролей 6. Применение паттернов проектирования в прикладной разработке 7. Паттерн "декоратор" 8. Паттерн Builder для создания Mock-объектов в целях юнит-тестирования 9. Роутинг в микросервисах с использованием Ocelot 10. Хостинг приложений, настройка реверс-проксирования запросов

2.1.3. Рекомендации по оцениванию тестовых заданий обучающихся

Максимальное количество баллов (государственная оценка)	Критерии
5 (отлично)	Выставляется обучающемуся, если обучающийся представил 90-100% правильных ответов.

4 (хорошо)	выставляется обучающемуся, если обучающийся представил 75-89% правильных ответов.
3 (удовлетворительно)	выставляется обучающемуся, если обучающийся представил 60-74% правильных ответов.
2 и менее (неудовлетворительно)	выставляется обучающемуся, если обучающийся представил менее 59% правильных ответов.

ТИПОВЫЕ ТЕСТОВЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ УРОВНЯ КОМПЕТЕНЦИЙ

Раздел №1. Технологии создания приложений на платформе .NET

Темы 1.1 – 1.3

Тестовые задания

1. Когда вызываются статические конструкторы классов в C#?
 - a) "После каждого обращения к статическим полям, методам и свойствам"
 - b) "Статических конструкторов в C# нет"
 - c) "Один раз при первом создании экземпляра класса или при первом обращении к статическим членам класса"
 - d) "Строгий порядок вызова не определен"

2. Каким образом можно перехватить добавление и удаление делегата из события?
 - "Использовать ключевые слова get и set"
 - "Для этого существуют специальные ключевые слова add и remove"
 - "Такая возможность не предусмотрена"
 - "Переопределить операторы + и – для делегата"

3. Что произойдет при выполнении следующего кода? `int i = 5; object o = i; long j = (long)o;`
 - a) "Значение переменной j предсказать нельзя"
 - b) "Ошибок не произойдет. Переменная j будет иметь значение 5"
 - c) "Произойдет ошибка времени компиляции"
 - d) "Средой исполнения будет вызвано исключение `InvalidCastException`"

4. Выберите средство, которое предоставляет C# для условной компиляции
 - a) "Директива `#typedef`"

- b) "Директива #if"
- c) "Директива #elseif"
- d) "Директива #switch"

5. Выберите правильный вариант, в которых пространство имен System содержит пространство имен Customizer

- a) "namespace System::Customizer { }"
- b) "namespace System { namespace Customizer { } }"
- c) "Нельзя создавать собственные пространства имен в пространстве имен System"

6. Чтобы использовать unsafe код в приложении, необходимо ...

- a) "Пометить методы, где используется небезопасный код с помощью ключевого слова fixed"
- b) "Пометить методы, где используется небезопасный код с помощью ключевого слова unsafe"
- c) "Пометить методы, где используется небезопасный код атрибутом Unsafe"

7. Реализацией какого паттерна (шаблона проектирования) являются события в C#?

- a) "Декоратор (Decorator)"
- b) "Посетитель (Visitor)"
- c) "Шаблонный метод (Template Method)"
- d) "Издатель-подписчик (Publisher-Subscriber)"

8. Чем отличаются константы и доступные только для чтения поля?

- a) "Константы можно изменять, а доступные только для чтения поля нет"
- b) "Ничем не отличаются"
- c) "Константы инициализируются во время компиляции, доступные только для чтения поля — во время выполнения"
- d) "Доступные только для чтения поля инициализируются во время компиляции, константы — во время выполнения"

9. Элемент, который нельзя пометить атрибутом

- a) "Структуры"
- b) "Все перечисленное можно пометить атрибутом"
- c) "Классы"

d) "Методы"

10. Как называется технология, благодаря которой возможно взаимодействие управляемого кода (managed code) с Win32 API функциями и СОМ-объектами?

- a) "WebServices"
- b) "Remoting"
- c) "CodeDOM"
- d) "Interop"

11. Что произойдет при компиляции проекта, где используется класс, структура, интерфейс или перечисление, помеченное атрибутом `Obsolete`?

- a) "Будет выведено предупреждение о том, что данный тип устарел, но сборка будет создана"
- b) "Произойдет ошибка компиляции, и проект не будет собран"
- c) "Сборка будет создана, но при запуске произойдет ошибка времени выполнения"
- d) "Атрибут `Obsolete` никак не влияет на компиляцию"
- e) "Нет нужного варианта ответа"

12. Вариант возможности структуры (value-типов)

- a) "Множественное наследование"
- b) "Boxing и Unboxing"
- c) "Переопределение конструктора по умолчанию"
- d) "Наследование от классов"

КОМПЛЕКТ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (зачёт с оценкой)

№ п/п	Содержание оценочного средства (вопросы к зачёту)	Индекс оцениваемой компетенции или ее элементов
Раздел 1. Тема 1.1.		
1.	Вопросы 1-6	ПК-2 3-1, ОПК-7 3-1, ОПК-7 В-1
Раздел 1. Тема 1.2.		
2.	Вопросы 7-12	ПК-2 3-1, ОПК-7 3-4, ОПК-7 В-1, ПК-2 3-1
Раздел 1. Тема 1.3.		

3.	Вопросы 13-18	ПК-2 З-1, ОПК-7 З-3, ОПК-7 В-1, ПК-2 У-1
Раздел 2. Тема 2.1.		
4.	Вопросы 19-24	ПК-2 З-1, ОПК-7 З-1, ОПК-7 В-1, ПК-2 В-1
Раздел 2. Тема 2.2.		
5.	Вопросы 24-29	ПК-2 З-1, ОПК-7 З-1, ОПК-7 В-1, ПК-2 З-1
Раздел 2. Тема 2.3.		
6.	Вопросы 30-36	ПК-2 З-1, ОПК-7 З-4, ОПК-7 В-1, ПК-2 У-1

ВОПРОСЫ К ЗАЧЁТУ

1. Как в приложениях Windows Forms интерфейс отделён от бизнес-логики
2. Какие управляющие графические элементы вы знаете?
3. Как сделать элемент невидимым, и как недоступным?
4. С помощью какого элемента можно имитировать временные задержки?
5. Из каких основных файлов состоит проект Windows Forms?
6. Какова структура проекта WPF?
7. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.
8. Построение диаграмм.
9. Построение графиков функций.
10. Технология создания приложений для БД.
11. Обработка исключений. Основные классы исключений в C#.
12. Оценка сложности алгоритмов. O-нотация.
13. Модульные тесты. Внедрение тестирования.
14. Список List в C#
15. Словарь Dictionary в C#
16. Файлы и каталоги
17. Однонаправленные списки и действия с ними.
18. Двухнаправленные списки.
19. Кольцевые списки. Списки с ключом.
20. Стек.
21. Очередь.
22. Дек.
23. Двоичные деревья. Формирование дерева. Добавление элемента в дерево. Печать дерева. Способы обхода дерева.
24. Двоичные деревья. Исключение компонента из дерева.
25. Библиотека STL. Классы контейнеры в C++.
26. Работа с классом Vector библиотеки STL
27. Работа с классом Map библиотеки STL

28. Работа с классом Dictionary библиотеки STL
29. Работа с удаленной БД из приложения Window Forms
30. Подсистема Graphic Device Interface
31. System.Drawing и System.Drawing.Drawing2D.

РЕЦЕНЗИЯ
на РАБОЧУЮ ПРОГРАММУ И ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДИСЦИПЛИНЫ
Б1.О.11.01 «Программирование»

шифр дисциплины в учебном плане, наименование

Направление подготовки 09.03.03 Прикладная информатика

код, наименование

Профиль «Прикладная информатика в управлении корпоративными информационными системами»

Разработчики: Э. М. Лебезова, старший преподаватель

(ФИО, НПР, участвовавших в разработке РПУД с указанием должности)

Кафедра: Информационных технологий

Представленная на рецензию рабочая программа и фонд оценочных средств дисциплины «Программирование» разработаны в соответствии с Государственным образовательным стандартом высшего профессионального образования по направлению подготовки 09.03.03 Прикладная информатика (Приказ Министерства образования и науки Донецкой Народной Республики от 21.01.2016 г. № 32); Федеральным государственным образовательным стандартом высшего образования – бакалавриата по направлению подготовки 09.03.03 Прикладная информатика (Приказ Министерства образования и науки Российской Федерации от 19.09.2017 г. № 922 с изменениями).

Рабочая программа дисциплины содержит требования к уровню подготовки бакалавров по приобретению теоретических знаний и практических навыков, которые предъявляются к обучающимся данного направления подготовки в ходе изучения учебной дисциплины.

В рабочей программе сформулированы цели и задачи освоения дисциплины.

В разделе «Место дисциплины в структуре образовательной программы» указаны требования к предварительной подготовке обучающихся, а также последующие дисциплины, для которых изучение данной будет необходимым.

В рабочей программе прописаны компетенции обучающегося (ПК-2, ОПК-7), формируемые в результате освоения дисциплины.

Раздел «Структура и содержание дисциплины» содержит тематическое и почасовое распределение изучаемого материала по видам занятий, а также часы на самостоятельную работу.

В разделе «Фонд оценочных средств» указаны оценочные средства для текущего контроля и промежуточной аттестации обучающегося (фонд оценочных средств представлен отдельным элементом УМКД).

Компетенции по курсу, указанные в рабочей программе, полностью соответствуют учебному плану и матрице компетенций.

В рабочей программе дисциплины представлен также перечень материально-технического обеспечения для осуществления всех видов занятий, предусмотренных учебным планом.

Рабочая программа дисциплины имеет логически завершённую структуру, включает в себя все необходимые и приобретаемые в процессе изучения навыки и умения.

Учебно-методическое обеспечение дисциплины представлено в программе перечнем основной и дополнительной литературы, методических материалов, библиотечно-информационных ресурсов, что является достаточным для успешного владения дисциплиной.

Таким образом, рабочая программа дисциплины «Программирование» соответствует всем требованиям к реализации программы и может быть рекомендована к использованию.

Рецензент:

доцент, канд. экон. наук, доцент

(должность, регалии)

подпись

И.В. Стешенко

ФИО

МП

30.08.2022 г.

