

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 19.11.2024 15:20:32

Уникальный идентификатор:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00ad6b37b55d8e1e3dbd7c78

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ростовский государственный экономический университет (РИНХ)»

Финансово-экономический колледж



УТВЕРЖДАЮ

Директор

Р. А. Сычев

« 31 » 10 2023 г.

Рабочая программа МДК Управление проектами

Специальность

09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ

Форма обучения	очная
Часов по учебному плану	126
в том числе:	
аудиторные занятия	92
самостоятельная работа	29

Ростов-на-Дону
2023 г.

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	3 (2.1)		4 (2.2)		Итого	
	Неделя		20			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	16	16	40	40	56	56
Практические	16	16	20	20	36	36
Консультации			2	2	2	2
Итого ауд.	32	32	60	60	92	92
Контактная работа	32	32	62	62	94	94
Сам. работа			29	29	29	29
Часы на контроль			3	3	3	3
Итого	32	32	94	94	126	126

ОСНОВАНИЕ

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование (Приказ Министерства образования и науки Российской Федерации от 9 декабря 2016 г. № 1547)

Рабочая программа составлена по образовательной программе 09.02.07 Информационные системы и программирование для набора 2023 года

программа среднего профессионального образования

Учебный план утвержден учёным советом вуза от 29.08.2023 протокол № 1

Программу составил(и): Преподаватель, Новожилов А.Н.

Председатель ЦМК: Горелько Е.А.

Рассмотрено на заседании ЦМК от 30.08.2023 протокол № 1

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	Целью МДК.03.02 Управление проектами является подготовка студентов по фундаментальным вопросам технологии управления проектами организации с использованием современного программного обеспечения. В соответствии с поставленными целями преподавание дисциплины реализует следующие задачи: современную технологию и методологию управления проектом; основные признаки и типы проектов, характеристики проектов, функции управления проектами; место проектной технологии в жизненном цикле организации; современное программное обеспечение в области управления проектами.
-----	--

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ООП:	МДК
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Элементы высшей математики
2.1.2	Теория вероятностей и математическая статистика
2.1.3	Дискретная математика с элементами математической логики
2.1.4	Операционные системы и среды
2.1.5	Архитектура аппаратных средств
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Учебная практика УП.03
2.2.2	Производственная практика ПП.03
2.2.3	Квалификационный экзамен ПМ.03
2.2.4	Сертификация информационных систем
2.2.5	Управление и автоматизация баз данных

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

3.1 Знать

ПК-3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.

Технологии решения задачи планирования и контроля развития проекта.

Принятые стандарты обозначений в графических языках моделирования.

Типовые функциональные роли в коллективе разработчиков, правила совмещения ролей.

ПК-3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.

Принципы построения системы диаграмм деятельности программного проекта.

Приемы работы с инструментальными средами проектирования программных продуктов.

ПК-3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.

Основные методы сравнительного анализа программных продуктов и средств разработки.

Основные подходы к менеджменту программных продуктов.

Основные методы оценки бюджета, сроков и рисков разработки программ.

3.2 Уметь

ПК-3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.

Работать с проектной документацией, разработанной с использованием графических языков спецификаций.

ПК-3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.

Выполнять оптимизацию программного кода с использованием специализированных программных средств.

Использовать методы и технологии тестирования и ревьюирования кода и проектной документации.

ПК-3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.

Проводить сравнительный анализ программных продуктов.

Проводить сравнительный анализ средств разработки программных продуктов.

Разграничивать подходы к менеджменту программных проектов.

3.3 Владеть

ПК-3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.

Построением заданных моделей программного средства с помощью графического языка (обратное проектирование).

ПК-3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.

Оптимизацией программного кода с использованием специализированных программных средств.

Навыками использования основных методологий процессов разработки программного обеспечения.

ПК-3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.

Навыками обоснования выбора методологии и средств разработки программного обеспечения.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Примечание
	Раздел 1. Инструменты для измерения характеристик и контроля качества и безопасности кода					
1.1	Программы, проекты и задачи. Экономические аспекты проекта /Лек/	3	2	ПК 3.1 ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.2	Формулировка идей проекта (сценариев развития проекта), выбор проекта методом количественных оценок /Пр/	3	2	ПК 3.1. ПК3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.3	Формулировка идей проекта (сценариев развития проекта), выбор проекта методом количественных оценок /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.4	Определение участников проекта, перечня этапов реализации проекта, а также материальных, трудовых ресурсов». /Пр/	3	2	ПК 3.1. ПК.3.3. ПК3.4.	Л1.1Л2.1 Э1 Э2	
1.5	Корректность программ /Лек/	3	2	ПК 3.1. ПК 3.3. ПК3.4.	Л1.1Л2.1 Э1 Э2	
1.6	Эталоны проверки корректности /Лек/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.7	Методы проверки корректности /Лек/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.8	Оценка экономической эффективности ИТ-проекта». /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.9	Метрики /Лек/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.10	Направления применения метрик /Лек/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.11	Метрики, направления применения метрик /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.12	Метрики сложности /Лек/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.13	Метрики стилистики /Лек/	3	2	ПК 3.1.	Л1.1Л2.1 Э1 Э2	

				ПК 3.3. ПК 3.4.	
1.14	Использование метрик стилистики /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.15	Использование метрик программного продукта /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.16	Использование метрик программного продукта /Пр/	3	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.17	Программный код /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.18	Исследование программного кода на предмет ошибок /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.19	Исследование программного кода на предмет ошибок /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.20	Исследование программного кода на предмет отклонения от алгоритма /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.21	Проверка целостности программного кода /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.22	Проверка целостности программного кода /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.23	Программные измерительные мониторы /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.24	Проверка целостности программного кода /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.25	Проверка целостности программного кода /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.26	Программные измерительные мониторы /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.27	Программные измерительные мониторы /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.28	Программные измерительные мониторы /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.29	Программные измерительные мониторы /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.30	Программные измерительные мониторы /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.31	Применение отладчиков и дизассемблера /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.32	Применение отладчиков и дизассемблера /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.33	Применение отладчиков и дизассемблера /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.34	Применение отладчиков и дизассемблера /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2

1.35	Применение отладчиков и дизассемблера /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.36	Применение отладчиков и дизассемблера /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.37	Применение отладчиков и дизассемблера (напримерOllyDbg, WinDbg, IdaPro) /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.38	Применение отладчиков и дизассемблера (напримерOllyDbg, WinDbg, IdaPro) /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.39	Применение отладчиков и дизассемблера (напримерOllyDbg, WinDbg, IdaPro) /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.40	Применение отладчиков и дизассемблера (напримерOllyDbg, WinDbg, IdaPro) /Конс/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.41	Применение отладчиков и дизассемблера (напримерOllyDbg, WinDbg, IdaPro) /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.42	Защита программ от исследования /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.43	Защита программ от исследования /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.44	Защита программ от исследования /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.45	Защита программ от исследования /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.46	Защита программ от исследования /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.47	Защита программ от исследования /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.48	Защита программ от исследования /Пр/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.49	Защита программ от исследования /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.50	Защита программ от исследования /Ср/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.51	Исследование кода вредоносных программ /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.52	Исследование кода вредоносных программ /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.53	Исследование кода вредоносных программ /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.54	Исследование кода вредоносных программ /Лек/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.55	Исследование кода вредоносных программ /Ср/	4	1	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2	
1.56	Выполнение измерений характеристик кода в среде VisualStudio /Пр/	4	2	ПК 3.1. ПК 3.3.	Л1.1Л2.1 Э1 Э2	

				ПК 3.4.	
1.57	Выполнение измерений характеристик кода в среде VisualStudio /Cp/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.58	Выполнение измерений характеристик кода в среде (например, Eclipse C/C++ и др.) /Cp/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.59	Выполнение измерений характеристик кода в среде (например, Eclipse C/C++ и др.) /Cp/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.60	Выполнение измерений характеристик кода в среде (например, Eclipse C/C++ и др.) /Cp/	4	2	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.61	Применение отладчиков и дизассемблера /Cp/	4	4	ПК 3.1. П ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2
1.62	Экзамен	4	3	ПК 3.1. ПК 3.3. ПК 3.4.	Л1.1Л2.1 Э1 Э2

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена. Перечень вопросов к экзамену:

1. Методические основы технологий создания ПО.
2. Визуальное моделирование.
3. Методы структурного анализа и проектирования ПО.
4. Методы объективно-ориентированного анализа и проектирование ПО.
5. Язык UML.
6. Сопоставление и взаимосвязь структурного и объектно-ориентированного подходов.
7. Методы моделирования бизнес-процессов и спецификации требований.
8. Методы анализа и проектирования ПО.
9. Технологии создания программного обеспечения.
10. Внедрение ТС ПО в организации.
11. Оценка и выбор ТС ПО.
12. Выполнение пилотного проекта.
13. Практическое внедрение ТС ПО.
14. Примеры ТС ПО.
15. Технология Rational Unified Process (IBM Rational Software).
16. Технология Oracle. Технология Borland.
17. Основные понятия моделирования бизнес-процессов.
18. Структурный подход к моделированию бизнес-процессов.
19. Объектно-ориентированный подход.
20. Принципы объектного подхода.
21. CASE технологии.
22. Цели, принципы и приемы экстремального программирования.
23. Назначение, фазы и свойства рационального унифицированного процесса.
24. Методология IDEF0.
25. Структура языка UML: сущности, отношения, диаграммы.
26. Типы сущностей в UML.
27. Виды отношений.
28. Правила языка UML.
29. Диаграммы UML.
30. Качество ИС, дефектологические свойства ИС.
31. Модель классификации критериев качества информационных систем.
32. Метрики кода программного обеспечения: назначение, виды.
33. Количественные метрики: показатели, типы.
34. Метрики сложности потока управления программ: показатели, типы.
35. Метрики сложности потока управления данными: показатели, типы.
36. Объектно-ориентированные метрики.
37. Метрики надежности. Гибридные метрики.
38. Тестирование ПО, этапы и принципы тестирования.
39. Методика тестирования программных систем.
40. Методы отладки.

Критерии оценивания:

5 баллов выставляется студентам за полный и правильный ответ на все вопросы билета с логическим обоснованием аргументов, в ответе нет ошибок.

4 балла выставляется студентам, если вопросы билета раскрыты полностью, но обоснования доказательства недостаточны, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя.

3 балла ставится студентам за правильный ответ на вопросы билета, при этом допущено более одной ошибки по изложению фактов или более двух-трех недочетов в ответе.

2 балла ставится студентам, если допущены существенные ошибки, показавшие, что обучающийся не обладает обязательными умениями по данной теме в полной мере.

5.2. Фонд оценочных средств для проведения текущего контроля

Представлен в Приложении 1 к рабочей программе МДК

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**6.1. Рекомендуемая литература****6.1.1. Основная литература**

	Авторы, составители	Заглавие	Издательство, год	Количество
Л1.1	Черткова Е. А.	Программная инженерия. Визуальное моделирование программных систем: Учебник для СПО: текст электронный	Юрайт, 2022	https://urait.ru/bcode/493226 неограниченный доступ для зарегистрированных пользователей

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Количество
Л2.1	Грекул В. И.	Проектирование информационных систем: учебник и практикум для среднего профессионального образования	Юрайт, 2022	https://urait.ru/bcode/496196 неограниченный доступ зарегистрированным пользователям

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Э1	Научная электронная библиотека eLIBRARY.RU https://elibrary.ru/defaultx.asp
Э2	Электронно-библиотечная система Znanium.com (Электронно-библиотечная система Znanium)

6.3. Перечень программного обеспечения

6.3.1	Операционная система - РЕД ОС «Стандартная» для Рабочих станций.
6.3.2	Офисный пакет - LibreOffice
6.3.3	Интернет-браузер - Chromium

6.4 Перечень информационных справочных систем

6.4.1	ИСС «КонсультантПлюс»
6.4.2	ИСС «Гарант»

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения
-----	---

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе МДК

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

МДК 03.02 Управление проектами

1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

1.1. Показатели и критерии оценивания компетенций:

УУД, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
ПК-3.1 Осуществлять ревьюирование программного кода в соответствии с технической документацией.			
Знать: Технологии решения задачи планирования и контроля развития проекта. Принятые стандарты обозначений в графических языках моделирования. Типовые функциональные роли в коллективе разработчиков, правила совмещения ролей.	Получение систематических знаний формулирует и знает понятия проекта информационных систем	Уровень знаний полнота и содержательность ответа, умение приводить примеры	Т(1-20), ПЗ (1-9)
Уметь: Работать с проектной документацией, разработанной с использованием графических языков спецификаций.	Сформировать систематическое умение выполнять задания, отвечать на вопросы, применять методы управления проектами	Уровень умения полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	Т(1-20), ПЗ (1-9)
Владеть: Построением заданных моделей программного средства с помощью графического языка (обратное проектирование).	Сформировать систематическое владение выполняет задания, проводит анализ данных и их обработку с использованием методов управления проектами	Уровень владения полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ (1-9)
ПК-3.3 Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.			
Знать: Принципы построения системы диаграмм деятельности программного проекта. Приемы работы с инструментальными средами проектирования программных продуктов.	Получение систематических знаний формулирует и знает понятия проекта информационных систем	Уровень знаний полнота и содержательность ответа умение приводить примеры	Т(1-20), ПЗ (1-9)
Уметь: Выполнять оптимизацию программного	Сформировать систематическое	Уровень умения полнота и	Т(1-20), ПЗ (1-9)

кода с использованием специализированных программных средств. Использовать методы и технологии тестирования и ревьюирования кода и проектной документации.	умение выполняет задания, отвечает на вопросы, применяет методы управления проектами	содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	
Владеть: Оптимизацией программного кода с использованием специализированных программных средств. Навыками использования основных методологий процессов разработки программного обеспечения.	Сформировать систематическое владение выполняет задания, проводит анализ данных и их обработку с использованием методов управления проектами	Уровень владения полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ (1-9)
ПК-3.4 Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.			
Знать: Основные методы сравнительного анализа программных продуктов и средств разработки. Основные подходы к менеджменту программных продуктов. Основные методы оценки бюджета, сроков и рисков разработки программ.	Получение систематических знаний формулирует и знает понятия проекта информационных систем	Уровень знаний полнота и содержательность ответа, умение приводить примеры	Т(1-20), ПЗ (1-9)
Уметь: Проводить сравнительный анализ программных продуктов. Проводить сравнительный анализ средств разработки программных продуктов. Разграничивать подходы к менеджменту программных проектов.	Сформировать систематическое умение выполняет задания, отвечает на вопросы, применяет методы управления проектами	Уровень умения полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	Т(1-20), ПЗ (1-9)
Владеть: Навыками обосновывать выбор методологии и средств разработки программного обеспечения.	Сформировать систематическое владение выполняет задания, проводит анализ данных и их обработку с использованием методов управления проектами	Уровень владения полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ (1-9)

Т – тестовые задания, ПЗ – практические задания.

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Тестовые задания:

1 семестр

1. Под термином «проект» следует понимать:

1) предприятие, у которого принципиальные отличия в условиях деятельности, целях, временных и материальных затратах, качественной характеристике продукции и специфике организации трудовой деятельности от других подобных организаций;

2) предпринимаемое усилие, которое формирует и сопровождает человеческие, материальные и финансовые ресурсы для осуществления успешных изменений с ограничением во времени и ресурсах, чтобы создать уникальный предмет работы заданной спецификации;

3) система уникальных действий, у которой имеются этапы начала и завершения и которая осуществляется организацией или отдельным человеком для решения задач с определенными параметрами, в отведенное время, с выделенными ресурсами для выполнения.

4) все ответы верные.

2. Что не относится к основным признакам проекта?:

1) наличие цели.

2) изменения.

3) инновационность.

4) ограничения во времени.

3. ИТ-проекты являются проектами с:

1) низкими рисками.

2) приемлемыми рисками.

3) высокими рисками.

4) допустимыми рисками.

4. К долгосрочным относятся проекты со сроком исполнения.

1) более 1 года.

2) более 2 лет.

3) более 3 лет.

4) более 5 лет.

5. Бизнес-аналитик, который входит в группу анализа, занимающейся извлечением, документированием и сопровождением требований к продукции:

1) отвечает за построение модели предметной области;

2) разрабатывает общее видение продукта, его интерфейс, поведение и ограничения (концепция продукции);

3) переводит требования к продукту в функциональные требования;

4) документирует и сопровождает требования к продукции.

6. Кто в группе анализа, которая занимается извлечением, документированием и сопровождением требований к продукции, занимается переводом требований к продукту в функциональные требования?

1) бизнес-аналитик;

2) бизнес-архитектор;

3) системный аналитик;

4) специалист по требованиям.

7. Что является наименьшей структурной единицей ИТ-проекта?

- 1) комплекс;
- 2) программа;
- 3) подпроект;
- 4) модуль.

8. Какой план устанавливает цели этапов, точки контроля, сроки реализации комплекса работ, взаимодействие исполнителей, потребности ресурсов на различных этапах?

- 1) концептуальный;
- 2) стратегический;
- 3) текущий;
- 4) оперативный.

9. Различают следующие виды окружения ИТ-проекта?

- 1) основное и вспомогательное;
- 2) техническое и программное;
- 3) собственное и заимствованное;
- 4) внутреннее и внешнее.

2 семестр

10. К ближнему окружению относятся?

- 1) участники проекта;
- 2) техническая платформа;
- 3) программная реализация;
- 4) требования бизнес-процесса.

11. Кто определяет основные требования к проекту, обеспечивает его ресурсами, привлекает исполнителей и взаимодействует с ними на протяжении всего времени существования ИТ-проекта?

- 1) руководитель проекта;
- 2) исполнитель;
- 3) заказчик;
- 4) системный аналитик.

12. Какой вклад в разработку стратегии вносит руководитель ИТ-службы?

- 1) оценка развивающихся технологий, требуемой инфраструктуры;
- 2) определение ИТ услуги, необходимые для бизнес-единиц;
- 3) оценка возможностей и потребностей;
- 4) выработка тактики для поддержки планов бизнеса.

13. Кто проектирует взаимодействие системы с пользователями на высоком уровне, определяет тенденцию, стиль по одному или более проектам?

- 1) технический эксперт;
- 2) юзабилити;
- 3) проектировщик взаимодействия;
- 4) архитектор.

14. Каким документом определяется персональный состав команды проекта?

- 1) приказом руководителя;
- 2) уставом проекта;
- 3) штатным расписанием;
- 4) договором.

15. Что происходит со штатным расписание предприятия при реализации ИТ-проекта?

- 1) остается неизменным;
- 2) пересматривается в стороны сокращения;
- 3) пересматривается в сторону увеличения;
- 4) временно изменяется.

16. Какую структуру управления рекомендуется выбирать при реализации крупных и сложных, а также нескольких ИТ-проектов?

- 1) матричную;
- 2) функциональную;
- 3) проектную;
- 4) не имеет значения.

17. Какая группа процессов управления проектом объединяет имеющиеся ресурсы для выполнения плана управления проектом?

- 1) группа инициации;
- 2) группа планирования;
- 3) группа исполнения;
- 4) группа мониторинга и управления.

18. К основным группам процессов управления проектом относятся:

- 1) стратегическое планирование;
- 2) управление финансами;
- 3) управление кадрами;
- 4) маркетинг.

19. К основным группам процессов управления проектом относятся:

- 1) производство;
- 2) продажа;
- 3) управление информационными технологиями;
- 4) обслуживание.

20. Как учитывается при оценке качества проекта «неключевая» функциональность?

- 1) учитывается полностью;
- 2) не учитывается полностью;
- 3) учитывается частично;
- 4) учитывается особым образом.

Критерии оценивания:

- оценка 5 («отлично») выставляется, если правильные ответы даны на 85- 100% вопросов

- оценка 4 («хорошо») выставляется студенту, если правильные ответы даны на 65-84% вопросов

- оценка 3 («удовлетворительно») выставляется студенту, если правильные ответы даны на 50-64% вопросов

- оценка 2 («неудовлетворительно») выставляется студенту, если правильные ответы даны на менее 50% вопросов

Практические задания:

1 семестр

№ 1

Тема: «Использование метрик программного продукта»

Метрика программного обеспечения (англ. software metric) — это мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Поскольку количественные методы хорошо зарекомендовали себя в других областях, многие теоретики и практики информатики пытались перенести данный подход и в разработку программного обеспечения. Как сказал Том ДеМарко, «вы не можете контролировать то, что не можете измерить». Набор используемых метрик может включать в себя: - количество строк кода; - цикломатическая сложность; - анализ функциональных точек; - количество ошибок на 1000 строк кода; - степень покрытия кода тестированием; - покрытие требований; - количество классов и интерфейсов; - метрики программного пакета от Роберта Сесиль Мартина; - связность; - порядок роста (имеется в виду анализ алгоритмов в терминах асимптотического анализа и O-нотации) и др.

Задание

заполнить таблицу примеров элементов данных для различных характеристик

Информационная характеристика	Элементы данных
Внешние Вводы	
Внешние выводы	
Внешние запросы	

Заполнить таблицу Правила учета элементов данных из графического интерфейса пользователя

Элементы данных	Правило учета
Группа радиокнопок	
Группа флажков (переключателей)	
Командные кнопки	
Списки	

Контрольные вопросы

1 Дать определение:

– Внешний интерфейсный файл

– Тип элемента-записи

– Тип элемента данных

2 Для каких продуктов используются метрики указателей свойств

3 Перечислите системные параметры приложения

№ 2

Тема: «Проверка целостности программного кода»

Краткие теоретические сведения:

Контроль целостности кода заключается в проверке состояния кода исполняемых модулей. Можно выделить следующие виды контроля целостности кода:

контроль запуска (загрузки) кода; контроль времени выполнения; комбинированный.

Для проверки целостности используются следующие методы контроля:

сравнение данных
сравнение значений хеш-функций;
установка защиты на память, содержащую код; проверка целостности потока выполнения

Задание:

1. Определить неизменяемые части исполняемых модулей
- 2 разработать модель и алгоритм контроля целостности кода

Контрольные вопросы:

- 1 Дать характеристику методу проверки целостности исполняемых модулей, основанному на сравнении значений хеш-функции.
- 2 какие задачи необходимо решить для достижения цели обеспечения предотвращения загрузки неизвестного или модифицированного кода, а также проверки целостности

№ 3

Тема: «Анализ потоков данных».

Краткие теоретические сведения

В основе модели лежат понятия внешней сущности, процесса, хранилища (накопителя) данных потока данных.

Поток данных представляет собой процесс передачи некоторой информации от источника к приемнику. Физически процесс передачи информации может происходить по кабелям под управлением программы или программной системы, а также вручную при участии устройств или людей вне проектируемой системы.

Задание

Разработать иерархию диаграмм потоков данных системы учета успеваемости студентов. В качестве внешних сущностей для системы выступают Декан, Заместитель декана по курсу и Сотрудник деканата. Определяем потоки данных между этими сущностями и системой.

Декан должен получать:

- сводку успеваемости по факультету (процент успеваемости групп, курсов и в целом по факультету) на текущий или указанный момент времени;
- полные сведения об учебе конкретного студента (успеваемость по всем изученным предметам всех завершенных семестров обучения с учетом пересдач).

Заместитель декана по курсу должен получать:

- сводку успеваемости по курсу (процент успеваемости по группам) на текущий или указанный момент;
- сведения о сдаче экзаменов и зачетов указанной группой; текущие сведения об успеваемости конкретного студента;
- полные сведения об учебе конкретного студента (успеваемость по всем изученным предметам всех завершенных семестров обучения с учетом пересдач);
- список задолжников по факультету, с указанием несданных предметов, и групп.

Сотрудник деканата должен обеспечивать:

- ввод списков студентов, зачисленных на первый курс;
- корректировку списков студентов в соответствии с приказами о зачислении, отчислении, переводе и т.п.;
- ввод учебных планов кафедр; ввод расписания сессии;
- ввод результатов сдачи зачетов и экзаменов на основании ведомостей и направлений.

Кроме того, сотрудник деканата должен иметь возможность получать:

– справку о прослушанных студентом предметах с указанием часов и итоговых оценок;
приложение к диплому выпускника также с указанием часов и итоговых оценок.

Контрольные вопросы:

- 1 Что такое Внешняя сущность
- 2 Что такое Процесс
- 3 Что такое Хранилище данных
- 4 Что такое Поток данных
- 5 Что иллюстрирует Диаграмма

№ 4

Тема «Использование метрик стилистики»

Наиболее простой метрикой стилистики и понятности является оценка уровня комментированности программы F:

$$F = N_{\text{ком}}/N_{\text{стр}},$$

где $N_{\text{ком}}$ - количество комментариев в программе;

$N_{\text{стр}}$ – количество строк или операторов исходного текста.

Таким образом, метрика F отражает насыщенность программы комментариями.

Исходя из практического опыта принято считать, что $F \geq 0.1$, т.е. на каждые десять строк программы должен приходиться минимум один комментарий. Как показывают исследования, комментарии распределяются по тексту программы неравномерно: в начале программы их избыток, а в середине или в конце – недостаток. Это объясняется тем, что в начале программы, как правило, расположены операторы описания идентификаторов, требующие более “плотного” комментирования. Кроме того, в начале программы также расположены “шапки”, содержащие общие сведения об исполнителе, характере, функциональном назначении программы и т.п. Такая насыщенность компенсирует недостаток комментариев в теле программы, и поэтому приведенная формула недостаточно точно отражает комментированность функциональной части текста программы.

Более удачен вариант, когда вся программа разбивается на n равных сегментов и для каждого из них определяется F_i :

$$F_i = \text{sign}(N_{\text{ком}}/N_{\text{стр}} - 0.1), \text{ при этом}$$

Уровень комментированности программы считается нормальным, если выполняется условие: $F = n$. В противном случае какой-либо фрагмент программы дополняется комментариями до нормального уровня.

Необходимо подчеркнуть, что стилистика и понятность программ тесно связана и с размером и со сложностью программ. Поэтому не следует забывать о некоторой условности группировки метрик программ.

Задание

Выполнить индивидуальное задание, выданное преподавателем Контрольные вопросы

- 1 Для чего используются метрики Холстеда
- 2 Что необходимо выполнить для вычисления параметров метрики Холстеда
- 3 Для чего используются метрика Маккейба
- 4 Что необходимо выполнить для вычисления параметров метрики Маккейба

2 семестр

№ 5

Тема «Выполнение измерений характеристик кода в среде VisualStudio»

Краткие теоретические сведения

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения. Помимо стандартного редактора и отладчика, которые существуют в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автозавершения кода, графические конструкторы и многие другие функции для упрощения процесса разработки.

Задание

Написать программу «Калькулятор», выполняющую четыре основных арифметических действия

Контрольные вопросы:

- 1 Понятие программы
- 2 Принципы проектирования программы
- 3 Основы языка программирования высокого уровня
- 4 Современные инструментальные среды программирования и их использование
- 5

№ 6

Тема «Выполнение измерений характеристик кода в среде Eclipse C/C»

Краткие теоретические сведения

Eclipse (МФА: [iˈklɪps], с англ. — «затмение»[8]) — свободная интегрированная среда разработки модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Eclipse служит в первую очередь платформой для разработки расширений, чем он и завоевал популярность: любой разработчик может расширить Eclipse своими модулями. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT), разрабатываемые инженерами QNX совместно с IBM, и средства для языков Ada (GNATbench, Hibachi), COBOL, FORTRAN, PHP, X10 (X10DT) и пр. от различных разработчиков. Множество расширений дополняет среду Eclipse диспетчерами для работы с базами данных, серверами приложений и др.

Задание

Написать программу «Калькулятор», выполняющую четыре основных арифметических действия

Контрольные вопросы:

- 1 Понятие программы
- 2 Принципы проектирования программы
- 3 Основы языка программирования высокого уровня
- 4 Современные инструментальные среды программирования и их использование

№ 7

Тема «Выполнение измерений характеристик кода в среде JavaDevelopmentKit»

Краткие теоретические сведения

Java Development Kit (сокращенно JDK) — бесплатно распространяемый компанией Oracle Corporation (ранее Sun Microsystems) комплект разработчика приложений на языке Java, включающий в себя компилятор Java (javac), стандартные библиотеки классов Java, примеры, документацию, различные утилиты и исполнительную систему Java (JRE). В состав JDK не

входит интегрированная среда разработки на Java, поэтому разработчик, использующий только JDK, вынужден использовать внешний текстовый редактор и компилировать свои программы, используя утилиты командной строки.

Задание

Написать программу «Калькулятор», выполняющую четыре основных арифметических действия

Контрольные вопросы:

- 1 Понятие программы
- 2 Принципы проектирования программы
- 3 Основы языка программирования высокого уровня
- 4 Современные инструментальные среды программирования и их использование

№ 8

Тема «Выполнение измерений характеристик кода в среде NetBeans»

Краткие теоретические сведения

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада[3] и ряда других.

Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org.

Последние версии NetBeans IDE поддерживают рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету и множество predefined шаблонов кода.

Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Среда разработки NetBeans по-умолчанию поддерживала разработку для платформ J2SE и J2EE. Начиная с версии 6.0 NetBeans поддерживает разработку для мобильных платформ J2ME, C++ (только g++) и PHP без установки дополнительных компонентов.

Задание

Написать программу «Калькулятор», выполняющую четыре основных арифметических действия

Контрольные вопросы:

- 1 Понятие программы
- 2 Принципы проектирования программы
- 3 Основы языка программирования высокого уровня
- 4 Современные инструментальные среды программирования и их использование

№ 9

Тема «Выполнение измерений характеристик кода в среде Development Tools»

Краткие теоретические сведения

PHP Development Tools — интегрированная среда разработки приложений на языке программирования PHP, разработанная на основе Eclipse, распространяемая на условиях лицензии Eclipse Public License.

Основные преимущества EDT заключаются в следующем:

- вы запускаете один экземпляр EDT и используете его для разработки многих

конфигураций, расширений конфигураций, внешних обработок и внешних отчетов. Быстро переключаетесь между разрабатываемыми приложениями внутри самой среды разработки: вы используете разные версии платформы и разные информационные базы для запуска и отладки одной и той же конфигурации;

- групповая разработка и версионирование выполняются с помощью внешней системы контроля версий Git;
- вы управляете информационными базами и публикациями на веб-серверах прямо из среды разработки;
- разработка мобильных приложений стала удобнее с использованием встроенного веб- сервера для их отладки и с возможностью выбора устройств для их запуска;
- проверка конфигурации выполняется «на лету», прямо в процессе изменения ее структуры и в процессе редактирования модулей. Найденные ошибки и возможные неполадки EDT показывает удобным образом;
- возможности редактора встроенного языка значительно выросли. Новую функциональность получили редактор запросов и редактор шаблонов ограничения доступа к данным;
- улучшены подходы к редактированию модулей, используются новые подходы: интроспекция модели и reflection (анализ модели объектов);
- инструмент быстрого исправления ошибок; контекстная подсказка с новыми возможностями; шаблоны генерации синтаксических конструкций.

Задание

Написать программу «Калькулятор», выполняющую четыре основных арифметических действия

Контрольные вопросы:

- 1 Понятие программы
- 2 Принципы проектирования программы
- 3 Основные преимущества EDT
- 4 Современные инструментальные среды программирования и их использование.

Критерии оценивания:

- оценка 5 («отлично») выставляется, если правильные ответы даны на 85- 100% вопросов
- оценка 4 («хорошо») выставляется студенту, если правильные ответы даны на 65-84% вопросов
- оценка 3 («удовлетворительно») выставляется студенту, если правильные ответы даны на 50-64% вопросов
- оценка 2 («неудовлетворительно») выставляется студенту, если правильные ответы даны на менее 50% вопросов

3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций состоит из текущего контроля.

Текущий контроль успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации и учитываются при оценивании знаний, умений, навыков и (или) опыта деятельности.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

МДК 03.02 Управление проектами

Методические указания для студентов по освоению МДК являются частью рабочей программы МДК (РПД) (приложением к рабочей программе).

РПД – рабочая программа, утвержденная директором колледжа для изучения МДК. Она определяет цели и задачи МДК, формируемые в ходе ее изучения компетенции и их компоненты, содержание изучаемого материала, виды занятий и объем выделяемого учебного времени, а также порядок изучения и преподавания МДК.

Для самостоятельной учебной работы студента важное значение имеют разделы «Структура и содержание дисциплины (модуля)» и «Учебно-методическое и информационное обеспечение дисциплины (модуля)». В первом указываются разделы и темы изучаемой МДК, а также виды занятий и планируемый объем (в академических часах), во втором – рекомендуемая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Для подготовки к текущему контролю студенты могут воспользоваться оценочными средствами, представленными в Приложении 1 к рабочей программе МДК.

1. Описание последовательности действий студента

Приступая к изучению МДК необходимо в первую очередь ознакомиться с содержанием РПД, где в разделе «Структура и содержание дисциплины (модуля)» приведено общее распределение часов аудиторных занятий и самостоятельной работы по темам МДК.

Залогом успешного освоения МДК является регулярное посещение занятий и выполнение предусмотренных программой заданий. Пропуск одного, а тем более нескольких занятий может осложнить освоение разделов курса.

Лекции имеют целью дать систематизированные основы научных знаний по содержанию МДК. При изучении и проработке теоретического материала необходимо:

- повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;
- при самостоятельном изучении теоретической темы подготовить конспект, используя рекомендованные в РПД литературные источники и электронные образовательные ресурсы.

Практические занятия проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы с учебной литературой.

В процессе практического занятия, как вида учебных занятий, обучающиеся выполняют одно или несколько практических заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

2. Самостоятельная работа студента

Самостоятельная работа студента – самостоятельная учебная деятельность студента, организуемая колледжем и осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

- воспитание самостоятельности, как личностного качества будущего специалиста.

Самостоятельная работа студента по МДК выполняется:

- самостоятельно вне расписания учебных занятий;
- с использованием современных образовательных технологий;
- работа со специальной литературой для подготовки к тестовым, практическим заданиям.

3. Рекомендации по работе с литературой и источниками

Работу с литературой следует начинать с анализа рабочей программы МДК, содержащей список основной и дополнительной литературы, а также знакомства с учебно-методическими разработками.

В случае возникновения затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным.

Работа с литературой не только полезна как средство более глубокого изучения МДК, но и является неотъемлемой частью профессиональной деятельности будущего выпускника.