



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Кубанский государственный университет»
в г. Славянске-на-Кубани

УТВЕРЖДАЮ



Проректор по работе с филиалами
ФГБОУ ВО «Кубанский
государственный университет»

А.А. Евдокимов

«15» декабря 2022 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

**МДК.02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

специальность 09.02.07 Информационные системы и программирование

Краснодар 2022

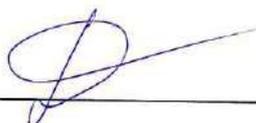
ЛИСТ

согласования рабочей программы по учебной дисциплине
МДК.02.02 «Инструментальные средства разработки программного обеспечения»

Специальность среднего профессионального образования:
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Нач. УМО филиала



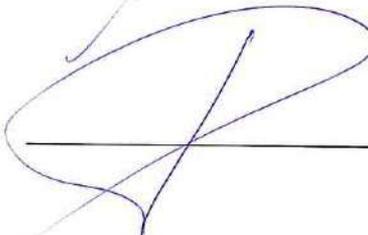
А.С. Демченко
«25» ноября 2022 г.

Заведующая библиотекой филиала



М.В. Фуфалько
«25» ноября 2022 г.

Нач. ИВЦ (программно-
информационное обеспечение
образовательной программы)



В.А. Ткаченко
«25» ноября 2022 г.

СОДЕРЖАНИЕ

1	ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
1.1	Область применения программы	5
1.2	Место учебной дисциплины в структуре программы подготовки специалистов среднего звена	5
1.3	Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины	5
1.4	Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)	5
2	СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	8
2.1	Объем учебной дисциплины и виды учебной работы	8
2.2	Структура дисциплины	8
2.3	Тематический план и содержание учебной дисциплины МДК.02.02 Инструментальные средства разработки программного обеспечения	8
2.4	Содержание разделов дисциплины	10
2.4.1	Занятия лекционного типа	10
2.4.2	Занятия семинарского типа	10
2.4.3	Практические занятия (Лабораторные занятия)	10
2.4.4	Содержание самостоятельной работы (Примерная тематика рефератов)	11
2.4.5	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	11
3	ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	13
3.1	Образовательные технологии при проведении лекций	13
3.2	Образовательные технологии при проведении практических занятий	13
4	УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ	14
4.1	Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	14
4.2	Перечень необходимого программного обеспечения	14
5	ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	15
5.1	Основная литература	15
5.2	Дополнительная литература	15
5.3	Периодические издания	15
5.4	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	16
6	МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ	18
7	КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	19
7.1	Паспорт фонда оценочных средств	19
7.2	Критерии оценки знаний	19
7.3	Оценочные средства для проведения текущей аттестации	21
7.4	Оценочные средства для проведения промежуточной аттестации	30
7.4.1	Примерные вопросы для проведения промежуточной аттестации	30
7.4.2	Примерные задачи для проведения промежуточной аттестации	32
8	ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	34

1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1 Область применения программы

Рабочая программа учебной дисциплины МДК.02.02 «Инструментальные средства разработки программного обеспечения» является частью основной профессиональной образовательной программой в соответствии с Федеральным государственным образовательным стандартом среднего профессионального образования (далее ФГОС СПО) и примерной основной образовательной программой для специальности 09.02.07 Информационные системы и программирование.

1.2 Место учебной дисциплины в структуре основной профессиональной образовательной программы

Дисциплина «Инструментальные средства разработки программного обеспечения» относится к общепрофессиональным дисциплинам профессиональной подготовки.

1.3 Цели и задачи учебной дисциплины. Требования к результатам освоения учебной дисциплины

В результате освоения дисциплины обучающийся должен иметь практический опыт:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения

В результате освоения дисциплины обучающийся должен уметь:

- использовать выбранную систему контроля версий;
- использовать методы для получения кода с заданной функциональностью и степенью качества

В результате освоения дисциплины обучающийся должен *знать*:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения

Максимальная учебная нагрузка обучающегося 108 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 98 часа;
- самостоятельная работа обучающегося 2 часа;
- консультации 2 часа;
- промежуточная аттестация 6 часов.

1.4. Перечень планируемых результатов обучения по дисциплине (Перечень формируемых компетенций)

Освоение дисциплины «Инструментальные средства разработки программного обеспечения» способствует формированию у студентов следующих профессиональных компетенций:

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	иметь практический опыт
1	ПК 2.2.	Выполнять интеграцию модулей в программное обеспечение	Основные подходы к интегрированию программных модулей. Основы верификации программного обеспечения. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций.	Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Использовать различные транспортные протоколы и стандарты форматирования сообщений.	Интегрировать модули в программное обеспечение. Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.
2	ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств.	Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации.	Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Использовать инструментальные средства отладки программных продуктов. Определять источники и приемники данных. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий.	Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

3	ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	<p>Модели процесса разработки программного обеспечения.</p> <p>Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей.</p> <p>Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации.</p> <p>Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов.</p> <p>Методы организации работы в команде разработчиков.</p>	<p>Использовать выбранную систему контроля версий.</p> <p>Использовать методы для получения кода с заданной функциональностью и степенью качества.</p> <p>Анализировать проектную и техническую документацию.</p> <p>Организовывать постобработку данных.</p> <p>Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций.</p>	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.
---	--------	--	---	--	---

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Всего часов	Семестр
		6
Учебная нагрузка (всего)	108	108
Аудиторная нагрузка (всего)	98	98
в том числе:		
лекционные занятия	54	54
практические занятия	44	44
Курсовое проектирование	–	–
Самостоятельная работа	2	2
Консультации	2	2
Промежуточная аттестация – экзамен	6	6

2.2 Структура дисциплины

Освоение учебной дисциплины МДК.02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ включает изучение следующих разделов и тем:

№	Тема	Всего часов	Лекции	Лабораторные	Самостоятельная работа
		100	54	44	2
1	<i>Тема 2.2.1 Современные технологии и инструменты интеграции.</i>	45	24	20	1
2	<i>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</i>	55	30	24	1

2.3 Тематический план и содержание учебной дисциплины

МДК.02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

<i>Наименование разделов и тем, междисциплинарных курсов (МДК)</i>	<i>Содержание учебного материала</i>	<i>Объем в часах</i>
<i>Тема 2.2.1 Современные технологии и инструменты интеграции.</i>	<i>Содержание</i>	<i>44</i>
	<i>1. Понятие репозитория проекта, структура проекта.</i>	<i>24</i>
	<i>2. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов.</i>	
	<i>3. Выбор источников и приемников Данных, сопоставление объектов данных.</i>	
	<i>4. Транспортные протоколы. Стандарты форматирования сообщений.</i>	
	<i>5. Организация работы команды в системе контроля версий.</i>	
	<i>В том числе практических занятий и лабораторных работ</i>	<i>20</i>

	1. Лабораторная работа «Разработка структуры проекта»	2
	2. Лабораторная работа «Разработка модульной структуры проекта (диаграммы модулей)»	2
	3. Лабораторная работа «Разработка перечня артефактов и протоколов проекта»	2
	4.Лабораторная работа «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)»	4
	5.Лабораторная работа «Разработка и интеграция модулей проекта (командная работа)»	4
	6.Лабораторная работа «Отладка отдельных модулей программного проекта»	4
	7.Лабораторная работа «Организация обработки исключений»	2
	Самостоятельная работа	1
Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств	Содержание	54
	1. Отладка программных продуктов. Инструменты отладки. Отладочные классы.	30
	2. Ручное и автоматизированное тестирование. Методы и средства организации тестирования.	
	3. Инструментарии анализа качества программных продуктов в среде разработки.	
	4.Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок.	
	5. Выявление ошибок системных компонентов.	
	В том числе практических занятий и лабораторных работ	24
	1. Лабораторная работа «Применение отладочных классов в проекте»	2
	2. Лабораторная работа «Отладка проекта»	2
	3. Лабораторная работа «Инспекция кода модулей проекта»	4
	4. Лабораторная работа «Тестирование интерфейса пользователя средствами инструментальной среды разработки»	4
	5. Лабораторная работа «Разработка тестовых модулей проекта для тестирования отдельных модулей»	4
	6. Лабораторная работа «Выполнение функционального тестирования»	4
	7. Лабораторная работа «Тестирование интеграции»	2
8. Лабораторная работа «Документирование результатов тестирования»	2	
	Самостоятельная работа	1
	Консультация	2
	Промежуточная аттестация	6
	Итого	108

2.4 Содержание разделов дисциплины

2.4.1 Занятия лекционного типа

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Тема 2.2.1 Современные технологии и инструменты интеграции.	<ol style="list-style-type: none"> 1. Понятие репозитория проекта, структура проекта. 2. Виды, цели и уровни интеграции программных модулей. Автоматизация бизнес-процессов. 3. Выбор источников и приемников Данных, сопоставление объектов Данных. 4. Транспортные протоколы. Стандарты форматирования сообщений. 5. Организация работы команды в системе контроля версий. 	У, КР
2	Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств	<ol style="list-style-type: none"> 1. Отладка программных продуктов. Инструменты отладки. Отладочные классы. 2. Ручное и автоматизированное тестирование. Методы и средства организации тестирования. 3. Инструментарии анализа качества программных продуктов в среде разработки. 4. Обработка исключительных ситуаций. Методы и способы идентификации сбоев и ошибок. 5. Выявление ошибок системных компонентов. 	У, КР

Примечание: Р - написание реферата, У - устный опрос, КР - контрольная работа

2.4.2 Занятия семинарского типа

- не предусмотрены

2.4.3 Практические занятия (Лабораторные занятия)

№ п/п	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Тема 2.2.1 Современные технологии и инструменты интеграции.	<ol style="list-style-type: none"> 1. Лабораторная работа «Разработка структуры проекта» 2. Лабораторная работа «Разработка модульной структуры проекта (диаграммы модулей)» 3. Лабораторная работа «Разработка перечня артефактов и протоколов проекта» 4. Лабораторная работа «Настройка работы системы контроля версий (типов импортируемых файлов, путей, фильтров и др. параметров импорта в репозиторий)» 5. Лабораторная работа «Разработка и интеграция модулей проекта (командная работа)» 	ЛР

		<p>6.Лабораторная работа «Отладка отдельных модулей программного проекта»</p> <p>7.Лабораторная работа «Организация обработки исключений»</p>	
2	<p>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</p>	<p>1. Лабораторная работа «Применение отладочных классов в проекте»</p> <p>2. Лабораторная работа «Отладка проекта»</p> <p>3. Лабораторная работа «Инспекция кода модулей проекта»</p> <p>4. Лабораторная работа «Тестирование интерфейса пользователя средствами инструментальной среды разработки»</p> <p>5. Лабораторная работа «Разработка тестовых модулей проекта Для тестирования отдельных модулей»</p> <p>6. Лабораторная работа «Выполнение функционального тестирования»</p> <p>7. Лабораторная работа «Тестирование интеграции»</p> <p>8. Лабораторная работа «Документирование результатов тестирования»</p>	ПР

Примечание: ПР- практическая работа

2.4.4 Содержание самостоятельной работы

- Подготовка к тестированию

2.4.5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа учащихся является важнейшей формой учебновоспитательного процесса.

Основная цель самостоятельной работы при изучении дисциплины - закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области архитектуры аппаратных средств.

НА САМОСТОЯТЕЛЬНУЮ РАБОТУ СТУДЕНТОВ ОТВОДИТСЯ 2 ЧАСОВ УЧЕБНОГО ВРЕМЕНИ.

Наименование раздела, темы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
<p>Тема 2.2.1 Современные технологии и инструменты интеграции.</p>	<p>Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/472502.</p>
<p>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</p>	<p>Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/472502.</p>

Кроме перечисленных источников учащийся может воспользоваться поисковыми системами сети Интернет по теме самостоятельной работы.

Для освоения дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе может быть использовано следующее учебно-методическое обеспечение:

- методические рекомендации к выполнению лабораторных работ;
- методические рекомендации к самостоятельной работе.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Учащийся должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

3 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для освоения курса «Технология разработки программного обеспечения» предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе обучения применяются образовательные технологии личностно-деятельностного, развивающего и проблемного обучения. Обязателен лабораторный практикум по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное тестирование, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

Тема	Виды применяемых образовательных технологий	Кол. час
<i>Тема 2.2.1 Современные технологии и инструменты интеграции.</i>	Аудиовизуальная технология, технология развивающего обучения	24
<i>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</i>	Аудиовизуальная технология, личностно-деятельностное обучение	30
	Итого	54
	В том числе интерактивное	54

3.2. Образовательные технологии при проведении практических занятий

Тема	Виды применяемых образовательных технологий	Кол. час
<i>Тема 2.2.1 Современные технологии и инструменты интеграции.</i>	Технология личностно-деятельностного обучения, технология проблемного обучения	20
<i>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</i>	Технология личностно-деятельностного обучения, технология проблемного обучения	24
	Итого	44
	В том числе интерактивное	44

4 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ МДК.02.01 ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

4.1 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Лаборатории *Программного обеспечения и сопровождения компьютерных систем*, оснащенные в соответствии с п. 6.1.2.1. Примерной программы по *специальности*

Оснащенные базы практики, в соответствии с п 6.1.2.3 Примерной программы по *специальности*.

4.2 Перечень необходимого программного обеспечения

1. 7-zip (лицензия на англ. <http://www.7-zip.org/license.txt>)
2. Adobe Acrobat Reader (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
3. Adobe Flash Player (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
4. Apache Open Office (лицензия - <http://www.openoffice.org/license.html>)
5. Free Commander (лицензия - <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>)
6. Google Chrome (лицензия - https://www.google.ru/chrome/browser/privacy/eula_text.html)
7. LibreOffice (в свободном доступе)
8. Mozilla Firefox (лицензия - <https://www.mozilla.org/en-US/MPL/2.0/>)
9. Oracle VM VirtualBox (лицензия - <https://www.virtualbox.org/wiki/GPL>)

5 ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

5.1 Основная литература

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1794453>. – Режим доступа: по подписке.

2. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2021. — 147 с. — (Профессиональное образование). — ISBN 978-5-534-09823-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473307>

5.2 Дополнительная литература

1. Гниденко, И. Г. Технология разработки программного обеспечения : учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472502>.

5.3 Периодические издания

- Computerworld Россия. – URL: <http://dlib.eastview.com/browse/publication/64081/udb/2071>.
- Windows IT Pro / Re. – URL: <http://dlib.eastview.com/browse/publication/64079/udb/2071>.
- БИТ. Бизнес & информационные технологии – URL : <http://dlib.eastview.com/browse/publication/66752/udb/2071>.
- Вестник Московского Университета. Серия 15. Вычислительная математика и кибернетика. - URL: <https://dlib.eastview.com/browse/publication/9166>.
- Вестник Санкт-Петербургского университета. Прикладная математика. Информатика. Процессы управления. URL: <https://dlib.eastview.com/browse/publication/71227/udb/2630>.
- Виртуализация. Облачные структуры. Системы хранения данных. – URL : <https://dlib.eastview.com/browse/publication/84826/udb/2071>.
- Журнал сетевых решений LAN. – URL: <http://dlib.eastview.com/browse/publication/64078/udb/2071>.
- Защита персональных данных. – URL : <https://dlib.eastview.com/browse/publication/90727/udb/2071>.
- Информатика и образование. - URL: <http://dlib.eastview.com/browse/publication/18946/udb/1270>.
- Информатика, вычислительная техника и инженерное образование. - URL: https://www.elibrary.ru/title_about.asp?id=32586.
- Информационно-управляющие системы. – URL: <http://dlib.eastview.com/browse/publication/71235/udb/2071>.

- Мир больших данных (Big Data). – URL :
<https://dlib.eastview.com/browse/publication/90728/udb/2071>.
- Мир ПК. – URL: [http://dlib.eastview.com/browse/publication/64067/udb/2071.\(2016\)](http://dlib.eastview.com/browse/publication/64067/udb/2071.(2016))
- Новые информационные технологии в автоматизированных системах
https://elibrary.ru/title_about.asp?id=32949.
- Открытые системы. СУБД. – URL:
<http://dlib.eastview.com/browse/publication/64072/udb/2071>.
- Прикладная информатика. – URL: https://e.lanbook.com/journal/2067#journal_name.
- Проблемы передачи информации. – URL:
http://www.mathnet.ru/php/archive.phtml?jrnid=ppi&wshow=contents&option_lang=rus.
- Программные продукты и системы. – URL:
<https://dlib.eastview.com/browse/publication/64086/udb/2071>.
- САПР и графика. - URL: <https://sapr.ru/list>,
- Системный администратор. – URL:
<https://dlib.eastview.com/browse/publication/66751/udb/2071>.
- Системный анализ и прикладная информатика. – URL:
https://e.lanbook.com/journal/2420#journal_name.
- Управление проектами и программами. – URL : <https://grebennikon.ru/journal-20.html#volume2019-3>.

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «BOOK.ru» [учебные издания – коллекция для СПО] : сайт. – URL: <https://www.book.ru/cat/576>.
2. ЭБС «Университетская библиотека ONLINE» [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы; мультимедийная коллекция, карты, онлайн-энциклопедии, словари] : сайт. – URL: http://biblioclub.ru/index.php?page=main_ub_red.
3. ЭБС издательства «Лань» [учебные, научные издания, первоисточники, художественные произведения различных издательств; журналы] : сайт. – URL: <http://e.lanbook.com>.
4. ЭБС «Юрайт» [учебники и учебные пособия издательства «Юрайт»] : сайт. – URL: <https://urait.ru/>.
5. ЭБС «Znaniium.com» [учебные, научные, научно-популярные материалы различных издательств, журналы] : сайт. – URL: <http://znaniium.com/>.
6. Научная электронная библиотека. Монографии, изданные в издательстве Российской Академии Естествознания [полнотекстовый ресурс свободного доступа] : сайт. – URL: <https://www.monographies.ru/>.
7. Научная электронная библиотека статей и публикаций «eLibrary.ru» [российский информационно-аналитический портал в области науки, технологии, медицины, образования; большая часть изданий – свободного доступа] : сайт. – URL: <http://elibrary.ru>.
8. Базы данных компании «Ист Вью» [периодические издания (на русском языке)] : сайт. – URL: <http://dlib.eastview.com>.
9. Российская электронная школа : государственная образовательная платформа [полный школьный курс уроков] : сайт. – URL: <https://resh.edu.ru/>.
10. Единое окно доступа к образовательным ресурсам : федеральная информационная система свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для всех уровней образования: дошкольное, общее, среднее профессиональное, высшее, дополнительное : сайт. – URL: <http://window.edu.ru>.
11. Федеральный центр информационно-образовательных ресурсов [для общего, среднего профессионального, дополнительного образования; полнотекстовый ресурс

свободного доступа] : сайт. – URL: <http://fcior.edu.ru>.

12. Единая коллекция цифровых образовательных ресурсов [для преподавания и изучения учебных дисциплин начального общего, основного общего и среднего (полного) общего образования; полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://school-collection.edu.ru>.

13. Официальный интернет-портал правовой информации. Государственная система правовой информации [полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://publication.pravo.gov.ru>.

14. Кодексы и законы РФ. Правовая справочно-консультационная система [полнотекстовый ресурс свободного доступа] : сайт. – URL: <http://kodeks.systems.ru>.

15. ГРАМОТА.РУ : справочно-информационный интернет-портал : сайт. – URL: <http://www.gramota.ru>.

16. Энциклопедиум [Энциклопедии. Словари. Справочники : полнотекстовый ресурс свободного доступа] // ЭБС «Университетская библиотека ONLINE» : сайт. – URL: <http://enc.biblioclub.ru/>.

17. СЛОВАРИ.РУ. Лингвистика в Интернете : лингвистический портал : сайт. – URL: <http://slovari.ru/start.aspx?s=0&p=3050>.

18. Электронный каталог Кубанского государственного университета и филиалов. – URL: <http://212.192.134.46/MegaPro/Web/Home/About>.

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учащиеся для полноценного освоения курса «Инструментальные средства разработки программного обеспечения» должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа учащихся является важнейшей формой учебно- познавательного процесса. Цель заданий для самостоятельной работы - закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины «Инструментальные средства разработки программного обеспечения»; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Отчеты по практическим занятиям должны содержать полные ответы на поставленные задания, необходимые таблицы должны быть заполнены.

Общие правила выполнения письменных работ

На первом занятии студенты должны быть проинформированы о необходимости соблюдения норм академической этики и авторских прав в ходе обучения. В частности, предоставляются сведения:

- общая информация об авторских правах;
- правила цитирования;
- правила оформления ссылок;

Все имеющиеся в тексте сноски тщательно выверяются и снабжаются «адресами».

Недопустимо включать в свою работу выдержки из работ других авторов без указания на это, пересказывать чужую работу близко к тексту без отсылки к ней, использовать чужие идеи без указания первоисточников (это касается и информации, найденной в Интернете). Все случаи плагиата должны быть исключены.

Список использованной литературы должен включать все источники информации, изученные и проработанные студентом в процессе выполнения работы, и должен быть составлен в соответствии с ГОСТ Р 7.0.5-2008 «Библиографическая ссылка. Общие требования и правила».

7 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «МДК.02.02 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

7.1 Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Компетенции	Наименование оценочного средства
1.	<i>Тема 2.2.1 Современные технологии и инструменты интеграции</i>	ПК 2.2, ПК 2.3, ПК 2.5	Проверка конспектов, устный опрос,
2.	<i>Тема 2.2.2 Инструментарий тестирования и анализа качества программных средств</i>	ПК 2.2, ПК 2.3, ПК 2.5	Проверка конспектов, тест

7.2 Критерии оценки знаний

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе проведения практических работ, тестирования, собеседования по результатам выполнения лабораторных работ, а также решения задач, составления рабочих таблиц и подготовки сообщений к уроку. Знания студентов на практических занятиях оцениваются отметками «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Код и наименование профессиональных и общих компетенций, формируемых в рамках	Критерии оценки	Методы оценки
Раздел модуля 2 Средства разработки программного обеспечения		
ПК 2.2 Выполнять интеграцию модулей в программное обеспечение	<p>Оценка «отлично» - в системе контроля версий выбрана верная версия проекта, проанализирована его архитектура, архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций в том числе с созданием классов-исключений (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий.</p> <p>Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, транспортные протоколы и форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля и дополнительная обработка исключительных ситуаций (при необходимости); определены качественные показатели полученного проекта; результат интеграции сохранен в системе контроля версий.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по обеспечению интеграции заданного модуля в предложенный программный проект</p> <p>Защита отчетов по практическим и лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

	<p>Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта, его архитектура доработана для интеграции нового модуля; выбраны способы форматирования данных и организована их постобработка, форматы сообщений обновлены (при необходимости); выполнена отладка проекта с применением инструментальных средств среды; выполнена доработка модуля (при необходимости); результат интеграции сохранен в системе контроля версий.</p>	
<p>ПК 2.3 Выполнять отладку программного модуля с использованием специализированных программных средств</p>	<p>Оценка «отлично» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; проанализирована и сохранена отладочная информация; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в полном объеме; результаты отладки сохранены в системе контроля версий. Оценка «хорошо» - в системе контроля версий выбрана верная версия проекта; протестирована интеграция модулей проекта и выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий.</p> <p>Оценка «удовлетворительно» - в системе контроля версий выбрана верная версия проекта; выполнена отладка проекта с применением инструментальных средств среды; выполнена условная компиляция проекта в среде разработки; определены качественные показатели полученного проекта в достаточном объеме; результаты отладки сохранены в системе контроля версий.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по выполнению отладки программного модуля.</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>
<p>ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования</p>	<p>Оценка «отлично» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены все имеющиеся несоответствия стандартам в предложенном коде.</p> <p>Оценка «хорошо» - продемонстрировано знание стандартов кодирования более чем одного языка программирования, выявлены существенные имеющиеся несоответствия стандартам в предложенном коде.</p> <p>Оценка «удовлетворительно» - продемонстрировано знание стандартов кодирования языка программирования, выявлены некоторые несоответствия стандартам в предложенном коде.</p>	<p>Экзамен/зачет в форме собеседования: практическое задание по инспектированию программного кода</p> <p>Защита отчетов по практическим и лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p>

7.3 Оценочные средства для проведения текущей аттестации

- фронтальный опрос
- индивидуальный устный опрос
- письменный контроль
- тестирование по теоретическому материалу
- практическая (лабораторная) работа

Форма аттестации	Знания	Умения	Владения (навыки)	Личные качества студента	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические (лабораторные) работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	оценка умения различать некоторые понятия	Оценка навыков логического анализа и синтеза при сопоставлении некоторых понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

Контрольная работа. Контрольная работа является набором практических заданий и задач по темам изучаемой дисциплины, позволяющих формировать знания, а также умения обучающихся в области архитектуры аппаратных средств.

Примеры задач и вопросов к контрольной работе:

1. Перечислите необходимые инструментальные средства разработки программ
2. Определите часто используемые инструментальные средства разработки программ
3. Определите специализированные инструментальные средства разработки программ
4. Проанализируйте интегрированные среды разработки
5. Проанализируйте средства разработки программного обеспечения
6. Сформулируйте определение «разработка программ»
7. Сформулируйте три этапа разработки программ
8. Проанализируйте средства проектирования приложений
9. Проанализируйте средства реализации программного кода
10. Проанализируйте средства тестирования программ
11. Сформулируйте классы инструментальных средств разработки программ

12. Проанализируйте четыре категории инструментальных программ, применяемые при проектировании экспертных систем
13. Проанализируйте оболочки экспертных систем
14. Проанализируйте языки программирования высокого уровня
15. Проанализируйте особенности среды программирования, поддерживающей несколько парадигм
16. Опишите дополнительные модули
17. Проанализируйте языки инженерии знаний
18. Проанализируйте язык логического программирования Prolog
19. Проанализируйте язык функционального программирования Lisp
20. Проанализируйте средства автоматизации разработки экспертных систем
21. Проанализируйте общее программное обеспечение
22. Проанализируйте специальное программное обеспечение
23. Проанализируйте инструментальную систему технологию программирования
24. Проанализируйте четыре класса компьютерной поддержки инструментальных систем технологий программирования
25. Сформулируйте понятие «комплексность»
26. Опишите ориентированность на коллективную разработку
27. Сформулируйте понятие технологическая определенность
28. Дайте определение понятию «интегрированность»
29. Перечислите компоненты инструментальных систем технологий программирования
30. Опишите базу данных разработки
31. Опишите инструментарий
32. Опишите интерфейсы
33. Опишите общую архитектуру инструментальных систем технологий программирования
34. Опишите инструментальную систему поддержки проекта
35. Опишите языково-зависимую инструментальную систему
36. Опишите пользовательский интерфейс
37. Опишите схему организации взаимодействия компьютера и пользователя
38. Опишите процедурно-ориентированный подход к разработке интерфейсов
39. Опишите объектно-ориентированный подход к разработке интерфейсов
40. Перечислите типы интерфейсов
41. Опишите интерфейс-меню
42. Опишите интерфейсы со свободной навигацией
43. Перечислите критерии оценки интерфейса пользователем
44. Перечислите факторы появления Case-технологий
45. Дайте определение Case-технологии?
46. Перечислите компоненты интегрированного Case-средства
47. Проанализируйте классификацию по категориям Case-средств
48. Проанализируйте классификацию по типам Case-средств

49. Перечислите вспомогательные типы Case-средств
50. Опишите структурный подход к разработке ИС
51. Проанализируйте объектно-ориентированный подход к разработке ИС
52. Перечислите факторы, усложняющие определение возможного эффекта от использования Case-средств
53. Перечислите качества организации для успешного внедрения Case-средств
54. Перечислите проблемы использования Case-средств
55. Проанализируйте структурный системный анализ
56. Опишите диаграмму «сущность-связь»
57. Опишите диаграмму классов
58. Проанализируйте язык графического описания UML
59. Опишите диаграмму компонентов
60. Опишите диаграмму композитной структуры
61. Опишите диаграмму развёртывания
62. Опишите диаграмму объектов
63. Опишите диаграмму пакетов
64. Опишите диаграмму деятельности
65. Проанализируйте преимущества UML
66. Опишите IDEF
67. Опишите диаграмму переходов состояний
68. Сформулируйте методологию функционального моделирования ИС
69. Дайте состав функциональной модели
70. Проанализируйте иерархию диаграмм
71. Опишите типы связей между функциями
72. Дайте характеристику современных Case-средств
73. Опишите методологию ARIS
74. Опишите программный продукт ARIS Express
75. Перечислите основные элементы, используемые в нотации ARIS
76. Проанализируйте архитектура ARIS

Тест. Тест представляет собой систему стандартизированных заданий, позволяющих автоматизировать процедуру измерения уровня знаний обучающихся.

1. Программное средство, предназначенное для поддержки разработки других программ, называется -...
 - 1) аппаратным инструментом
 - 2) программным инструментом
 - 3) программной средой
 - 4) инструментарий технологии программирования
2. Анализаторы обеспечивают...
 - 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла

- 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
 - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
 - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
3. Преобразователи обеспечивают...
- 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
 - 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
 - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
 - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
4. Сколько классов инструментальных средств выделяют в инструментальной среде разработки и сопровождения программ?
- 1) 2
 - 2) 4
 - 3) 3
 - 4) 5
5. Среда программирования предназначена для...
- 1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла
 - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
 - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
 - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
6. Инструментальные среды программирования бывают
- 1) языково-ориентированные среды и среды общего назначения
 - 2) объектно-ориентированные и языково-ориентированные среды
 - 3) среды общего назначения и прикладные среды
 - 4) среды общего назначения, прикладные среды, логические и математические среды
7. Для поддержки разработки программного продукта на каком-либо одном языке программирования используют...
- 1) среду программирования общего назначения
 - 2) языково-ориентированную среду программирования
 - 3) интерпретирующую среду программирования
 - 4) прикладную среду программирования
8. Синтаксически-управляемая инструментальная среда программирования базируется на знании
- 1) семантики языка программирования
 - 2) синтаксиса языка программирования
 - 3) синтаксиса и семантики языка программирования
 - 4) основных управляющих структур языка программирования
9. На рисунке представлена классификация
- 1) инструментальной системы технологии программирования
 - 2) инструментальной среды разработки и сопровождения программ
 - 3) рабочего места компьютерной технологии
 - 4) языков программирования
10. Инструментальная система технологии программирования – это...

- 1) программное средство, предназначенное для поддержки разработки других программ
 - 2) устройство компьютера, специально предназначенное для поддержки разработки программного средства
 - 3) интегрированная совокупность программных и аппаратных инструментов, поддерживающая все процессы разработки и сопровождения больших программных продуктов
 - 4) логически связанная совокупность программных и аппаратных инструментов, поддерживающих разработку ПП
11. Устройство компьютера, специально предназначенное для поддержки разработки программного средства, называется -...
- 1) аппаратным инструментом
 - 2) программным инструментом
 - 3) программной средой
 - 4) инструментарий технологии программирования
12. Редакторы обеспечивают...
- 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
 - 2) автоматически приводить документы к другой форме представления или переводить документ одного вида к документу другого вида
 - 3) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
 - 4) выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
13. Инструменты, поддерживающие процесс выполнения программ, обеспечивают...
- 1) конструирование тех или иных программных продуктов и документов на различных этапах жизненного цикла
 - 2) автоматический привод документов к другой форме представления или перевод документа одного вида к документу другого вида
 - 3) возможность выполнять на компьютере описание процессов или отдельных их частей, представленных в виде, отличном от машинного кода
 - 4) статическую обработку документов, осуществляя различные виды их контроля, выявление определенных их свойств и накопление статистических данных, либо динамический анализ программ
14. Инструментальная система технологии программирования предназначена для...
- 1) поддержки всех процессов разработки и сопровождения в течение всего жизненного цикла ПС и ориентирована на коллективную разработку больших программных систем с длительным жизненным циклом
 - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
 - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
 - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
15. Рабочее место компьютерной технологии предназначено для...
- 1) конструирования тех или иных программных продуктов и документов на различных этапах жизненного цикла
 - 2) автоматического перевода документов к другой форме представления или перевода документа одного вида к документу другого вида
 - 3) поддержки ранних этапов разработки программ и автоматической генерации программ по спецификациям
 - 4) поддержки процессов программирования (кодирования), тестирования и отладки программ
16. Инструментальные среды программирования содержат
- 1) редактор, анализатор и компилятор
 - 2) редактор, интерпретатор и компилятор
 - 3) интерпретатор, компилятор, преобразователь

4) редактор и интерпретатор

17. Для поддержки разработки программного продукта на разных языках программирования (например, текстовый редактор, редактор связей или интерпретатор языка целевого компьютера) используют...

- 1) среду программирования общего назначения
- 2) языково-ориентированную среду программирования
- 3) интерпретирующую среду программирования
- 4) прикладную среду программирования

18. На рисунке представлена классификация

- 1) инструментальной системы технологии программирования
- 2) инструментальной среды программирования
- 3) рабочего места компьютерной технологии
- 4) языков программирования

19. При использовании компьютерных технологий для разработки ПП жизненный цикл ПП представлен следующей цепочкой:

- 1) прототипирование – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
- 2) прототипирование – разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
- 3) разработка спецификаций – автоматизированный контроль спецификаций – кодогенерация – комплексная отладка и тестирование – аттестация, применение, сопровождение
- 4) прототипирование – разработка спецификаций – кодогенерация – аттестация, применение, сопровождение

20. Основными чертами инструментальной системы технологии программирования являются...

- 1) массовость, дискретность, результативность, определенность, понятность
- 2) комплексность, ориентированность на коллективную разработку, технологическая определенность, интегрированность
- 3) актуальность, непротиворечивость, полнота
- 4) комплексность, актуальность, интегрированность, массовость, понятность

1. Современные крупные проекты информационных систем характеризуются следующими особенностями:

- 1) сложность описания, требующая тщательного моделирования и анализа данных и процессов
- 2) наличие совокупности тесно взаимодействующих компонентов
- 3) наличие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений
- 4) невозможность интеграции существующих и вновьразрабатываемых приложений;

2. Под CASE-средства понимаются программные средства, поддерживающие...

- 1) процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО
- 2) процессы тиражирования программного продукта
- 3) процессы создания и эксплуатации программного продукта
- 4) процессы компиляции и интерпретации программных продуктов

3. Репозиторий Case – средства – это...

- 1) специализированная база данных проекта, предназначенная для отображения состояния проектируемой системы в каждый момент времени
- 2) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
- 3) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
- 4) компонент, позволяющий получать информацию о проектах в виде отчетов
- 5) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта

- 6) набор системных утилит по обслуживанию репозитория
4. Графический редактор Case – средства – это...
- 1) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
 - 2) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
 - 3) компонент, позволяющий получать информацию о проектах в виде отчетов
 - 4) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
5. Верификатор Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
 - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
 - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
 - 4) набор системных утилит по обслуживанию репозитория
6. Документатор проекта Case – средства – это...
- 1) компонент, позволяющий получать информацию о проектах в виде отчетов
 - 2) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
 - 3) набор системных утилит по обслуживанию репозитория
 - 4) компонент, обеспечивающий создание и редактирование в интерактивном режиме элементов диаграмм и связей между ними
7. Сервис Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
 - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
 - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
 - 4) набор системных утилит по обслуживанию репозитория
8. Администратор проекта Case – средства – это...
- 1) компонент, служащий для контроля правильности построения диаграмм в заданной методологии проектирования
 - 2) компонент, позволяющий получать информацию о проектах в виде отчетов
 - 3) компонент, выполняющий запуск проекта, задание начальных параметров и назначение и изменение прав доступа к элементам проекта
 - 4) набор системных утилит по обслуживанию репозитория
9. Какие методологии проектирования используют Case – средства?
- 1) структурного и модульного проектирования
 - 2) структурного и объектно-ориентированного проектирования
 - 3) объектно-ориентированного и нисходящего проектирования
 - 4) нисходящего и восходящего проектирования
10. Структурное проектирование системы основано на...
- 1) объектно-ориентированной декомпозиции
 - 2) алгоритмической декомпозиции
 - 3) модульной декомпозиции
 - 4) функциональной декомпозиции
11. Объектно-ориентированное проектирование системы основано на...
- 1) объектно-ориентированной декомпозиции
 - 2) алгоритмической декомпозиции
 - 3) модульной декомпозиции
 - 4) функциональной декомпозиции
12. Case – средства представляют собой...
- 1) набор инструментальных средств для проектирования программного продукта

- 2) набор программных средств для сопровождения программного продукта
- 3) набор программных и инструментальных средств, поддерживающие все процессы жизненного цикла программного продукта
- 4) набор аппаратных средств, поддерживающих все процессы жизненного цикла программного продукта
13. Компания-разработчик приобрела новое Case – средство. Сразу ли компания получит ожидаемый результат от применения новой технологии?
- 1) да
- 2) нет
14. Сколько классов Case – средств выделяют?
- 1) 5
- 2) 3
- 3) 7
- 4) 2
15. Case – средства анализа и проектирования, предназначенные для
- 1) моделирования данных и генерации схем баз данных
- 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
- 3) обеспечения комплексной поддержки требований к создаваемой системе
- 4) поддержки всего жизненного цикла программного продукта
16. Case – средства управления требованиями предназначены для
- 1) моделирования данных и генерации схем баз данных
- 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
- 3) обеспечения комплексной поддержки требований к создаваемой системе
- 4) поддержки всего жизненного цикла программного продукта
17. Case – средства проектирования баз данных предназначены для
- 1) моделирования данных и генерации схем баз данных
- 2) построения и анализа моделей деятельности организаций (предметной области) или моделей проектируемой системы
- 3) обеспечения комплексной поддержки требований к создаваемой системе
- 4) поддержки всего жизненного цикла программного продукта
18. На каких стандартах базируется технология освоения и внедрения Case – средств?
- 1) ГОСТ 2107-99
- 2) IEEE (IEEE Std 1348-1995 и IEEE Std 1209-1992)
- 3) AES
- 4) ISO
19. Из каких этапов состоит процесс освоения и внедрения Case – средств?
- 1) определение потребностей в CASE-средствах, оценка и выбор CASE- средств, практическое внедрение CASE-средств
- 2) определение потребностей в CASE-средствах, оценка и выбор CASE- средств, выполнение пилотного проекта, практическое внедрение CASE- средств
- 3) определение потребностей в CASE-средствах, проектирования CASE-средств, практическое применение CASE-средств
- 4) проектирование CASE-средств, оценка и внедрение CASE-средств, практическое применение CASE-средств
20. Критериями для выбора CASE-средств могут являться
- 1) открытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, обеспечение целостности проекта, независимость от программно-аппаратной платформы и СУБД
- 2) модифицируемость, простота, эффективность, учет человеческого фактора, многоплатформенность
- 3) закрытая архитектура, поддержка полного жизненного цикла ИС с обеспечением эволюционности ее развития, простота, эффективность

4) максимальная зависимость от программных и аппаратных средств системы и характеристик самой системы, жесткая привязка к конкретным информационным процессам, прочность внутренней связи отдельных компонентов системы

21. Комплексность компьютерной поддержки разработки ПП с использованием инструментальной системы технологии программирования означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что все инструменты объединены единым пользовательским интерфейсом

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

22. Ориентированность инструментальной системы технологии программирования на коллективную разработку означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что все инструменты объединены единым пользовательским интерфейсом

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

23. Технологическая определенность инструментальной системы технологии программирования означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

24. Интегрированность инструментальной системы технологии программирования по данным означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что инструменты действуют в соответствии с фиксированной информационной схемой системы, определяющей зависимость различных используемых в системе фрагментов данных друг от друга

25. Интегрированность инструментальной системы технологии программирования по пользовательскому интерфейсу означает

1) что система технологии программирования охватывает все процессы разработки и сопровождения ПС и что продукция этих процессов согласована и взаимоувязана

2) что система технологии программирования должна поддерживать управление работой

коллектива и для разных членов этого коллектива обеспечивать разные права доступа к различным фрагментам продукции технологических процессов

3) что ее комплексность ограничивается рамками какой-либо конкретной технологии программирования

4) что все инструменты объединены единым пользовательским интерфейсом

7.4 Оценочные средства для проведения промежуточной аттестации

Форма аттестации	Знания	Умения	Владение (навыки)	Личные качества студента	Примеры оценочных средств
Итоговая аттестация					
Экзамен	Контроль знания базовых положений в области операционных систем	Оценка умения понимать специальную терминологию	Оценка навыков логического сопоставления и характеристики объектов, работы и администрирования операционной системы	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения решать типовые задачи в области операционных систем	Оценка навыков логического мышления при решении задач в области операционных систем	Оценка способности грамотно и четко излагать ход решения задач в области архитектуры операционных систем и аргументировать результаты	Задачи прилагаются

7.4.1 Примерные вопросы для проведения промежуточной аттестации

1. Перечислите необходимые инструментальные средства разработки программ
2. Определите часто используемые инструментальные средства разработки программ
3. Определите специализированные инструментальные средства разработки программ
4. Проанализируйте интегрированные среды разработки
5. Проанализируйте средства разработки программного обеспечения
6. Сформулируйте определение «разработка программ»
7. Сформулируйте три этапа разработки программ
8. Проанализируйте средства проектирования приложений
9. Проанализируйте средства реализации программного кода
10. Проанализируйте средства тестирования программ
11. Сформулируйте классы инструментальных средств разработки программ
12. Проанализируйте четыре категории инструментальных программ, применяемые при проектировании экспертных систем
13. Проанализируйте оболочки экспертных систем
14. Проанализируйте языки программирования высокого уровня

15. Проанализируйте особенности среды программирования, поддерживающей несколько парадигм
16. Опишите дополнительные модули
17. Проанализируйте языки инженерии знаний
18. Проанализируйте язык логического программирования Prolog
19. Проанализируйте язык функционального программирования Lisp
20. Проанализируйте средства автоматизации разработки экспертных систем
21. Проанализируйте общее программное обеспечение
22. Проанализируйте специальное программное обеспечение
23. Проанализируйте инструментальную систему технологию программирования
24. Проанализируйте четыре класса компьютерной поддержки инструментальных систем технологий программирования
25. Сформулируйте понятие «комплексность»
26. Опишите ориентированность на коллективную разработку
27. Сформулируйте понятие технологическая определенность
28. Дайте определение понятию «интегрированность»
29. Перечислите компоненты инструментальных систем технологий программирования
30. Опишите базу данных разработки
31. Опишите инструментарий
32. Опишите интерфейсы
33. Опишите общую архитектуру инструментальных систем технологий программирования
34. Опишите инструментальную систему поддержки проекта
35. Опишите языково-зависимую инструментальную систему
36. Опишите пользовательский интерфейс
37. Опишите схему организации взаимодействия компьютера и пользователя
38. Опишите процедурно-ориентированный подход к разработке интерфейсов
39. Опишите объектно-ориентированный подход к разработке интерфейсов
40. Перечислите типы интерфейсов
41. Опишите интерфейс-меню
42. Опишите интерфейсы со свободной навигацией
43. Перечислите критерии оценки интерфейса пользователем
44. Перечислите факторы появления Case-технологий
45. Дайте определение Case-технологии?
46. Перечислите компоненты интегрированного Case-средства
47. Проанализируйте классификацию по категориям Case-средств
48. Проанализируйте классификацию по типам Case-средств
49. Перечислите вспомогательные типы Case-средств
50. Опишите структурный подход к разработке ИС
51. Проанализируйте объектно-ориентированный подход к разработке ИС

52. Перечислите факторы, усложняющие определение возможного эффекта от использования Case-средств
53. Перечислите качества организации для успешного внедрения Case-средств
54. Перечислите проблемы использования Case-средств
55. Проанализируйте структурный системный анализ
56. Опишите диаграмму «сущность-связь»
57. Опишите диаграмму классов
58. Проанализируйте язык графического описания UML
59. Опишите диаграмму компонентов
60. Опишите диаграмму композитной структуры
61. Опишите диаграмму развёртывания
62. Опишите диаграмму объектов
63. Опишите диаграмму пакетов
64. Опишите диаграмму деятельности
65. Проанализируйте преимущества UML
66. Опишите IDEF
67. Опишите диаграмму переходов состояний
68. Сформулируйте методологию функционального моделирования ИС
69. Дайте состав функциональной модели
70. Проанализируйте иерархию диаграмм
71. Опишите типы связей между функциями
72. Дайте характеристику современных Case-средств
73. Опишите методологию ARIS
74. Опишите программный продукт ARIS Express
75. Перечислите основные элементы, используемые в нотации ARIS
76. Проанализируйте архитектура ARIS

7.4.2 Примерные задачи для проведения промежуточной аттестации

1. Анализ выбранного стиля программирования
2. Разработка проекта программного обеспечения
3. Разработка структурного алгоритма
4. Разработка программного продукта с использованием объектно-ориентированного программирования
5. Разработка справочной системы
6. Тестирование методом «белого ящика»
7. Тестирование методом «черного ящика»
8. Способы анализа граничных решений
9. Способы диаграмм причин-следствий
10. Нисходящее тестирование интеграций
11. Восходящее тестирование интеграций

12. Анализ предметной области
13. Автоматизированное тестирование
14. Отладка программ
15. Оптимизация программ
16. Работа в составе бригады

8 ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Понятие репозитория проекта, структура проекта

Репозиторий – хранилище информации, связанной с проектом разработки программного продукта в течение всего его жизненного цикла.

Большинство технологических подходов к разработке ПО предполагает работу с тремя основными типами информации – модельными спецификациями, интерфейсом прикладного программиста и окружением проекта. В соответствии с этими типами выделяют и три класса уровней репозитория:

1. Модельный.
2. Программного интерфейса.
3. Окружения.

Уровень моделирования достаточно хорошо может быть описан универсальным языком UML. Данный язык является абстрактным, не привязанным к конкретной модели. Язык дает возможность описать зависимости элементов, иерархию, взаимосвязи, свойства и т. п.

Уровень программного интерфейса разумно описывать с помощью языка определения интерфейсов IDL, обеспечивающего независимость спецификации интерфейсов от их реализации. Уровень играет не только роль промежуточного слоя – его средства также поддерживают распределенное программирование.

Репозиторий окружения программного проекта предназначен для хранения информации, разделяемой компонентами и подкомпонентами систем программирования в процессе их работы. При этом основными группами и подгруппами полезной информации считаются следующие:

1. Языково-независимая группа:
 - информация для отладчика;
 - информация для анализатора исходных текстов.
2. Языково-зависимая группа:
 - информация для шаблонов;
 - коды встроенных функций;
 - виртуальные функции.
3. Группа контроля репозитория:
 - контроль информации о версиях;
 - контроль отношения к проекту;
 - тип параллельной обработки;
 - тип управления репозиторием;
 - проверка цифровой подписи.

Главными достоинствами применения репозитория окружения являются:

- эффективность работы с информацией;
- использование информации для целей оптимизации;
- распределенность (из которой следуют доступность, параллелизм и специализация);

- модульность, включающая независимость от конкретных инструментов (например, компиляторов);

- возможность работы с репозиторием как в архитектуре «клиент-сервер», так и в «связанном» с инструментом режиме.

Статистика отмечает, что около 80% ПО создается по уже имеющемуся. Следовательно, необходимо иметь электронную библиотеку, которая будет поддерживать архивы и интеллектуальный поиск нужных прототипов и фрагментов. Одним из наиболее известных репозитория является Microsoft Repository.

2. Поддержка коллективной разработки:

системы управления версиями

Системы управления версиями (контроля версий) – программное обеспечение, предназначенное для отслеживания изменений между различными версиями файловых документов и разделения доступа к ним.

Такие системы наиболее широко применяются при разработке программного обеспечения – для хранения исходных кодов программ. Однако могут с успехом применяться и в других областях, в которых ведётся работа с большим количеством непрерывно изменяющихся электронных документов (например, в САПР).

Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости, возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение и т.д.

Большинство систем контроля версий используют централизованное хранилище документов, управляемое специальным сервером, который выполняет большую часть функций по управлению версиями. При передаче файлов под управление такой системой, она создает для них собственный репозиторий, с помощью которого отслеживает и хранит все изменения.

Системы контроля версий, как правило, отслеживают только изменения (дельты) между версиями файлов (а не сами полные версии), что позволяет эффективно использовать дисковое пространство.

Часто бывает, что над одним проектом одновременно работают несколько человек. Если два человека изменяют один и тот же файл, то один из них может случайно отменить изменения, сделанные другим. Системы управления версиями отслеживают такие конфликты и предлагают средства их решения. Большинство систем может автоматически объединить (слить) изменения, сделанные разными разработчиками. Однако, такое автоматическое объединение изменений, обычно, возможно только для текстовых файлов и при условии, что изменялись разные (непересекающиеся) части этого файла. Такое ограничение связано с тем, что большинство систем управления версиями ориентированы на поддержку процесса разработки программного обеспечения, а исходные коды программ хранятся в текстовых файлах. Если автоматическое объединение выполнить не удалось, система может предложить решить проблему вручную.

Часто выполнить слияние невозможно, ни в автоматическом, ни в ручном режиме, например, если формат файла неизвестен. Некоторые системы управления версиями дают возможность заблокировать файл в хранилище. Блокировка не

позволяет другим пользователям получить рабочую копию и обеспечивает, таким образом, исключительный доступ только тому пользователю, который работает с документом.

Многие системы управления версиями предоставляют ряд других возможностей:

1. Позволяют создавать разные варианты одного документа, т. е. ветки, с общей историей изменений до точки ветвления и с разными – после неё.

2. Дают возможность узнать, кто и когда добавил или изменил конкретную строку кода в файле.

3. Ведут журнал изменений, в который пользователи могут записывать информацию о том, что и почему они изменили в данной версии.

4. Контролируют права доступа пользователей, разрешая или запрещая чтение или изменение информации, в зависимости от того, кто запрашивает это действие.

На практике системы управления версиями рекомендуется использовать не только при коллективной, но и при индивидуальной (авторской) разработке ПО.

РЕЦЕНЗИЯ

на рабочую программу учебной дисциплины
МДК.02.02 Инструментальные средства разработки программного
обеспечения
для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины МДК.02.02 «Инструментальные средства разработки программного обеспечения» соответствует ФГОС специальности среднего профессионального образования 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от «09» декабря 2016 г. № 1547, зарегистрирован в Министерстве юстиции России 26.12.2016 г. рег. № 44936.

В рабочую программу учебной дисциплины включены разделы «Паспорт рабочей программы учебной дисциплины», «Структура и содержание учебной дисциплины», «Образовательные технологии», «Условия реализации программы учебной дисциплины», «Перечень основных и дополнительных информационных источников, необходимых для освоения дисциплины», «Методические рекомендации обучающимся по освоению дисциплины», «Оценочные средства для контроля успеваемости» и «Дополнительное обеспечение дисциплины».

Структура и содержание рабочей программы соответствуют целям образовательной программы СПО по специальности 09.02.07 «Информационные системы и программирование» и будущей профессиональной деятельности студента.

Объем рабочей программы учебной дисциплины полностью соответствует учебному плану подготовки по данной специальности. В программе четко сформулированы цели обучения, а также прогнозируемые результаты обучения по дисциплине.

На основании проведенной экспертизы можно сделать заключение, что рабочая программа учебной дисциплины МДК.02.02 «Инструментальные средства разработки программного обеспечения» по специальности 09.02.07 «Информационные системы и программирование» соответствует требованиям стандарта, профессиональным требованиям, а также современным требованиям рынка труда.

Инженер-программист 1 категории
отдела АСУТП управления АСУТП,
КИПиА, МОП Краснодарского РПУ
филиала «Макрорегион ЮГ» ООО ИК
«СИБИНТЕК»

« » 20 г.

ООО ИК «СИБИНТЕК»
Филиал «Макрорегион ЮГ»
352300, г. Тбилиси, ул. Солонная, 40
ИНН 7723013014 / КПП 772301001

М.В. Литус