

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ СОЦИАЛЬНЫЙ ИНСТИТУТ»

Утверждаю
Декан факультета
_____ Ж.В. Игнатенко
«18» мая 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО
ОБЕСПЕЧЕНИЯ

Специальность: 09.02.11 Разработка и управление программным обеспечением

Квалификация: Программист

Направленность: Разработка приложений для мобильных платформ

Форма обучения очная

Разработана
Канд.пед.наук, доцент
_____ Г.А. Бондарева

Согласована
зав. кафедрой ПИМ
_____ Д.Г. Ловянников

Рекомендована
на заседании кафедры
от «18» мая 2026 г.
протокол № 10
Зав. кафедрой _____ Д.Г. Ловянников

Одобрена
на заседании учебно-методической
комиссии факультета
от «18» мая 2026 г.
протокол № 9
Председатель УМК _____ Ж.В. Игнатенко

Ставрополь, 2026 г.

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	3
4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ	4
5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ	4
5.1. Содержание дисциплины.....	4
5.2. Структура дисциплины.....	5
5.3. Практические занятия и семинары	5
5.4. Лабораторные работы	6
5.5. Самостоятельное изучение разделов (тем) дисциплины.....	6
6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	6
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	6
8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	11
8.1. Основная литература.....	21
8.2. Дополнительная литература.....	22
8.3. Программное обеспечение	22
8.4. Базы данных, информационно-справочные и поисковые системы, Интернет- ресурсы	22
9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	23
10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	23

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины «Инструментальные средства проектирования программного обеспечения» формирование компетенций в области применения современных инструментальных средств для анализа требований, проектирования и разработки модулей программного обеспечения.

Задачи:

- Изучить основы стандартизации и управления требованиями к ПО.
- Освоить методы и средства (IDEF, DFD и др.) для анализа требований и проектирования.
- Сформировать практические навыки проектирования и разработки модулей ПО с использованием современных технологий.
- Научиться выбирать оптимальные инструментальные средства для решения профессиональных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «ОП.10. Инструментальные средства проектирования программного обеспечения» включена в обязательную часть общепрофессионального цикла образовательной программы и находится в логической и содержательно-методической связи с другими дисциплинами ОПОП.

Предшествующие дисциплины (курсы, модули, практики)	Последующие дисциплины (курсы, модули, практики)
1. Основы работы с информацией	1. ПМ.02 Разработка и интеграция модулей программного обеспечения 2. ПМ.03 Разработка приложений для мобильных платформ 3. ПМ.В.02 Проектирование и разработка веб-приложений 4. Стандартизация, сертификация и техническое документоведение 5. Управление ИТ-проектами

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций по данной специальности:

Компетенции	Результаты обучения
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	знать: <ul style="list-style-type: none">– основные принципы проектирования модулей программного обеспечения;– методы анализа требований и способов определения функциональности модуля; уметь: <ul style="list-style-type: none">– принципы создания интерфейсов для взаимодействия с другими модулями и системами– принципы обеспечения безопасности, производительности и масштабируемости при проектировании модулей.
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности	

ПК 2.1. Проектировать модули программного обеспечения.	<ul style="list-style-type: none"> – создавать архитектурные диаграммы и документацию; – анализировать требования к модулю и определять его функциональность; – разрабатывать модули программного обеспечения с использованием инструментальных средств и технологий.
ПК 2.2. Разрабатывать модули программного обеспечения.	

4. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Общий объем дисциплины составляет 60 академических часов.

Вид учебной работы	Всего часов	Семестр
		2* (4**)
Аудиторные занятия (работа обучающихся во взаимодействии с преподавателем) (всего)	40	40
в том числе:		
Лекции (Л)	20	20
Практические занятия (ПЗ)	20	20
Самостоятельная работа (всего) (СР)	20	20
в том числе:		
Самоподготовка (самостоятельное изучение разделов, проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам, рубежному контролю и т.д.)	20	20
Промежуточная аттестация (зачет)	6	6
Общий объем, час	66	66

* на базе среднего общего образования

** на базе основного общего образования

5. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

5.1. Содержание дисциплины

№ раздела (темы)	Наименование раздела (темы)	Содержание раздела (темы)
1	Современные технологии и инструменты проектирования	Современные принципы и методы разработки программных приложений. Методы организации работы в команде разработчиков. Системы контроля версий. Командная работа.
2	Инструментальные средства разработки программных продуктов	Инструментальные средства разработки программ. Классификация. Необходимое ПО; часто используемое ПО; специализированные интегрированные среды. Универсальные характеристики инструментальных средств. (фирма-производитель, автор; название продукта; номер последней версии; класс продукта, который установил для него производитель; тип дистрибуции программы и стоимость; наличие и тип поддержки, ее стоимость; доступность и

		качество документации; простота и понятность интерфейса; наличие пробных версий (для платных программ); сайт программы и возможность ее скачивания; размер дистрибутива и его состав.)
3	Описание и анализ требований. Язык моделирования UML	Описание требований: унифицированный язык моделирования - краткий словарь. Диаграммы UML. Описание и оформление требований (спецификация). Моделирование бизнес-процесса. Диаграммы стандарта IDEF0. Методология моделирования баз данных на основе модели «сущность-связь». Диаграммы стандарта IDEF1 и IDEF1X. Методология построения объектно-ориентированных систем. Диаграммы стандарта IDEF4.

5.2. Структура дисциплины

№ раздела (темы)	Наименование раздела (темы)	Всего	Количество часов				
			Л	ПЗ	С	ЛР	СР
1.	Современные технологии и инструменты проектирования	14	4	-	-	-	10
2.	Инструментальные средства разработки программных продуктов	16	6	8	-	-	2
3	Описание и анализ требований. Диаграммы IDEF	30	10	12			8
4.	Промежуточная аттестация	6					
	Общий объем, час	66	20	20	-	-	20

5.3 Практические занятия и семинары

№ п/п	№ раздела (темы)	Вид (ПЗ, С)	Тема	Количество часов
1	1	ПЗ	Разработка и интеграция модулей проекта (командная работа)	2
2	1	ПЗ	Разработка модульной структуры проекта (диаграммы модулей)	2
3	1	ПЗ	Анализ предметной области	2
4	3	ПЗ	Разработка и оформление технического задания	
5	3	ПЗ	Построение архитектуры программного средства. Изучение работы в системе контроля версий	2
6	3	ПЗ	Построение диаграммы Вариантов использования и диаграммы Последовательности	2
7	3	ПЗ	Построение диаграммы Кооперации и диаграммы Развертывания	2
8	3	ПЗ	Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов	2
9	3	ПЗ	Построение диаграммы компонентов	2
10	3	ПЗ	Построение диаграмм потоков данных	2
			Общий объем, час	20

5.4. Лабораторные работы

Лабораторные работы рабочим учебным планом не предусмотрены.

5.5. Самостоятельное изучение разделов (тем) дисциплины

№ раздела (темы)	Темы, выносимые на самостоятельное изучение	Количество часов
1	Современные цифровые ресурсы и технологии проектирования и разработки программного обеспечения	10
1	Инструментальные средства проектирования. Особенности установки, настройки и использования.	2
2	Унифицированный язык моделирования	8

6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основные технологии обучения:

- работа с информацией, в том числе с использованием ресурсов сети Интернет;
- подготовка и реализация проектов (мультимедийных презентаций и пр.) по заранее заданной теме;
- исследование конкретной темы и оформление результатов в виде доклада с презентацией;
- работа с текстами учебника, дополнительной литературой;
- выполнение индивидуальных заданий.

Информационные технологии:

- сбор, хранение, систематизация, обработка и представление учебной и научной информации;
- обработка различного рода информации с применением современных информационных технологий;
- самостоятельный поиск дополнительного учебного и научного материала, с использованием поисковых систем и сайтов сети Интернет, электронных энциклопедий и баз данных;
- использование электронной почты преподавателей и обучающихся для рассылки, переписки и обсуждения возникших учебных проблем;
- использование дистанционных образовательных технологий (при необходимости)

Активные и интерактивные образовательные технологии, используемые в аудиторных занятиях

№ раздела (темы)	Вид занятия (Л, ПЗ, С, ЛР)	Используемые интерактивные и активные образовательные технологии	Количество часов
1	Л	Лекция-дискуссия.	2
2	Л	Лекция-визуализация	2
3	ПЗ	Работа малыми группами	2

Практическая подготовка обучающихся в данной дисциплине не предусмотрена.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

7.1. Типовые задания для текущего контроля.

Перечень типовых контрольных вопросов для устного опроса

1. Что подразумевается под «современными принципами разработки ПО»? (Ожидается ответ про Agile/Scrum, CI/CD, DevOps).
2. Назовите ключевые преимущества гибких методологий (Agile) перед каскадной моделью (Waterfall).
3. Опишите основные роли и артефакты в методологии Scrum.
4. Как системы контроля версий (например, Git) помогают в организации командной работы?
5. Объясните разницу между централизованной (SVN) и распределенной (Git) системой контроля версий.
6. Что такое merge conflict в Git и как его разрешают?
7. По каким критериям можно классифицировать инструментальные средства разработки? (Ожидается упоминание коммерческих/бесплатных, специализированных/универсальных, языково-ориентированных и т.д.).
8. Что такое интегрированная среда разработки (IDE) и каковы ее основные преимущества перед простым текстовым редактором?
9. Назовите основные компоненты типичной IDE.
10. Какие универсальные характеристики необходимо оценить при выборе инструментального средства для проекта? (Ожидается ответ на основе вашего текста: стоимость, поддержка, документация и т.д.).
11. Приведите примеры специализированных сред разработки для разных языков программирования (например, Android Studio, PyCharm, Xcode).
12. Почему важно учитывать тип дистрибуции (проприетарная, open-source) и стоимость поддержки при выборе ПО для предприятия?
13. Какова основная цель языка моделирования UML?
14. Перечислите основные типы диаграмм UML и briefly опишите назначение каждой. (Достаточно 3-4 основных: Use Case, Class, Sequence, Activity).
15. Что такое «спецификация требований к ПО» (SRS) и какую роль в ее создании играют UML-диаграммы?
16. Для чего используется диаграмма вариантов использования (Use Case Diagram) и из каких основных элементов она состоит?
17. Опишите назначение и ключевые элементы диаграммы классов (Class Diagram).
18. В чем заключается основная цель методологии IDEF0 и для чего применяется контекстная диаграмма?
19. Чем отличается назначение диаграмм IDEF0 (функциональное моделирование) и IDEF1X (логическое моделирование данных)?
20. Опишите основные элементы нотации IDEF1X: сущности, атрибуты и виды связей (связи-идентификации и неидентификации).

21. Критерии и шкала оценивания устного опроса

отлично	1) студент полно излагает материал, дает правильное определение основных понятий; 2) обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; 3) излагает материал последовательно и правильно с точки зрения норм литературного языка.
хорошо	студент дает ответ, удовлетворяющий тем же требованиям, что и для отметки, но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.

удовлетворительно	студент обнаруживает знание и понимание основных положений данной темы, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
неудовлетворительно	студент обнаруживает незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал. Оценка «неудовлетворительно» отмечает такие недостатки в подготовке, которые являются серьезным препятствием к успешному овладению последующим материалом.

Перечень типовых заданий в тестовой форме

1. Какие программы можно отнести к системному программному обеспечению:

1. прикладные программы;
2. операционные системы;
3. игровые программы.

2. Какие программы можно отнести к системному ПО:

1. текстовые редакторы;
2. электронные таблицы;
3. драйверы;
4. графические редакторы.

3. Специфические особенности ПО как продукта:

1. низкие материальные затраты при создании программ;
2. возможность создание программ небольшие коллективом или даже одним человеком;
3. продажа по ценам ниже себестоимости (лицензирование);
4. разнообразие решаемых задач с помощью программных средств.

4. Какие программы можно отнести к системному ПО:

1. программа расчета заработной платы;
2. электронные таблицы;
3. СУБД (системы управления базами данных).

5. Какие программы нельзя отнести к системному ПО:

1. компиляторы языков программирования;
2. операционные системы;
3. игровые программы;
4. системы управления базами данных.

6. Какие программы можно отнести к прикладному программному обеспечению:

1. электронные таблицы;
2. таблицы решений;
3. СУБД (системы управления базами данных).

7. Какие программы можно отнести к прикладному ПО:

1. диспетчер программ;
2. программа расчета заработной платы;
3. программа «Проводник» (Explorer).

8. Какие программы нельзя отнести к прикладному ПО:

1. компиляторы и (или) интерпретаторы;

2. текстовые и (или) графические редакторы;
 3. электронные таблицы.
- 9. Этап, занимающий наибольшее время, в жизненном цикле программы:**
1. сопровождение;
 2. проектирование;
 3. тестирование;
 4. программирование;
 5. формулировка требований.
- 10. Этап, занимающий наибольшее время, при разработке программы:**
1. тестирование;
 2. сопровождение;
 3. проектирование;
 4. программирование;
 5. формулировка требований.
- 11. Первый этап в жизненном цикле программы:**
1. формулирование требований;
 2. анализ требований;
 3. проектирование;
 4. автономное тестирование;
 5. комплексное тестирование.
- 12. Один из необязательных этапов жизненного цикла программы:**
1. оптимизация;
 2. проектирование;
 3. тестирование;
 4. программирование;
 5. анализ требований.
- 13. Модель, представляющая собой объект, который ведет себя как реальный объект, но не выглядит как таковой — это**
1. физическая модель;
 2. аналоговая модель;
 3. типовая модель;
 4. математическая модель.
- 14. Модель, представляющая то, что исследуется с помощью увеличенного или уменьшенного описания объекта или системы — это**
1. аналитическая;
 2. физическая;
 3. типовая;
 4. математическая.
- 15. Какой из структурных элементов включает в себя процесс моделирования?**
5. анализ;
 6. модель;
 7. объект;
 8. субъект.
- 16. Автоматизация процесса управления не включает в себя**
1. этап анализа
 2. этап планирования и разработки
 3. этап управления ходом разработки
 4. нет правильного ответа
- 17. Математической моделью конфликтных ситуаций является:**
1. теория игр
 2. сетевая модель
 3. имитационная модель

4. транспортная модель
- 18. Моделирование основывается на принципах:**
- 1 Выбор модели оказывает определяющее влияние на подход к решению проблемы и на то, как будет выглядеть это решение
 - 2 Декомпозиции системы на отдельные подзадачи
 - 3 Децентрализации управления системой
 - 4 Каждая модель может быть представлена с различной степенью точности; лучшие модели – те, что ближе к реальности
 - 5 Нельзя ограничиваться созданием только одной модели. Наилучший подход при разработке любой нетривиальной системы – использовать совокупность нескольких моделей, почти независимых друг от друга
- 19. Какое моделирование предполагает представление модели в виде некоторого алгоритма - компьютерной программы?**
- 1 имитационное
 - 2 смешанное
 - 3 аналитическое
 - 4 физическое
- 20. Что является результатом методологии IDEF0?**
- 1 Функциональная модель;
 - 2 Реляционная таблица
 - 3 Модель IS- AS
 - 4 Модель IS- BE
- 21. Диаграмма IDEF0 может содержать следующие типы диаграмм:**
- 1 Диаграмму классов
 - 2 Контекстную диаграмму, диаграмму декомпозиции
 - 3 Диаграмму компонентов
 - 4 Диаграмму дерева узлов
 - 5 Диаграмму взаимодействий
 - 6 Диаграмму только для экспозиции (FEO)
 - 7 Диаграмму последовательности, диаграмму кооперации
- 22. Уровни логической модели:**
- 1 Диаграмма сущность
 - 2 Диаграмма связь
 - 3 Диаграмма сущность-связь
 - 4 Модель данных, основанная на ключах
 - 5 Полная операционная модель
 - 6 Полная атрибутивная модель
- 23. Модель DFD описывает:**
- 1 Функции обработки стрелок (arrow)
 - 2 Функции обработки информации (работы)
 - 3 Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации
 - 4 Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации
 - 5 Функции обработки внешних ссылок
 - 6 Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data stor+ E)
- 24. MS Visio позволяет создавать схемы, чертежи, диаграммы с помощью:**
- 1 Встроенных шаблонов
 - 2 Панели инструментов
 - 3 Трафаретов
 - 4 Графических редакторов

5 Панели рисования

6 Стандартных модулей

25. Язык UML – это:

1 Язык программирования высокого уровня

2 Язык для разработки систем искусственного интеллекта

3 Язык управления базами данных

4 Язык для визуализации, специфицирования, конструирования и документирования артефактов программных систем

5 Язык создания запросов в базах данных

Критерии и шкала оценки тестовых заданий

Количество правильных ответов	Оценка
86 – 100%	отлично
71 – 85%	хорошо
53 – 70%	удовлетворительно
52%	неудовлетворительно

Перечень типовых практических/ситуационных задач

Задание 1. Произведите анализ предметной области «Туристическое агентство».

1. Опишите бизнес-процессы предметной области.
2. Разработайте диаграммы Вариантов использования и Последовательности.
3. Разработайте регламент выполнения процесса «Работа с клиентами» в информационной системе для Туристического агентства.

Задание 2. Произведите анализ предметной области «Магазин оргтехники».

1. Опишите бизнес-процессы предметной области.
2. Постройте диаграммы Компонентов и Поток данных.
3. Разработайте регламент выполнения процесса «Поставки товара» в информационной системе для «Магазин оргтехники».

Задание 3. Произведите анализ предметной области «Фирмы по оказанию бухгалтерских услуг».

1. Опишите бизнес-процессы предметной области.
2. Разработайте диаграммы Развёртывания для веб-приложения и Композитной структуры.
3. Разработайте регламент выполнения процесса «Разработка документации по работе с клиентами» в информационной системе для Фирмы по оказанию бухгалтерских услуг.

Задание 4. Произведите анализ предметной области «Агрегатор авиабилетов» и на его основе:

1. Составьте техническое задание на разработку программного продукта .
2. Оформите диаграмму объектов предметной области.

Задание 5. С использованием инструментов контроля версий Git подготовить закрытый репозиторий для совместной разработки на 3 человека.

1. Подготовить и описать типовой файл README.md, продемонстрировать базовые функции форматирования.
2. Создать ветки для каждого участника команды.
3. Добавить участников в репозиторий с правами чтения и записи.

Задание 6. Произведите анализ предметной области «Салон сотовой связи» и на его основе:

1. Разработать диаграммы Состояний и Классов.
2. Оценить необходимое количество тестов и разработать тестовые сценарии.

Задание 7. Используя инструмент контроля версий Git, загрузить из удалённого репозитория проект. С использованием интегрированной среды разработки ПО Microsoft Visual Studio:

1. произвести тестирование трассировкой проекта на предмет поиска ошибок выполнения;
2. проанализировать встроенными средствами удобство обслуживания, сложность организации и взаимосвязь классов;
3. выполнить оптимизации программного кода и приведения к единому стандарту кодирования.

Критерии и шкала оценки практического задания/работы

отлично	студент самостоятельно и правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя понятия дисциплины.
хорошо	студент самостоятельно и в основном правильно решил учебно-профессиональную задачу, уверенно, логично, последовательно и аргументировано излагал свое решение, используя понятия дисциплины.
удовлетворительно	студент в основном решил учебно-профессиональную задачу, допустил несущественные ошибки, слабо аргументировал свое решение, используя в основном понятия дисциплины.
неудовлетворительно	ставится, если: студент не решил учебно-профессиональную задачу.

7.2. Типовые задания для промежуточной аттестации (дифференцированный зачет)

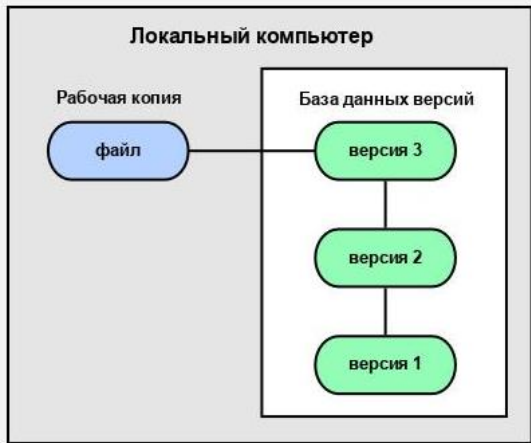
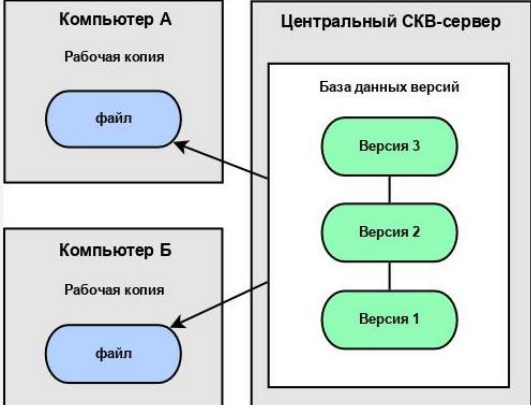
Перечень типовых контрольных вопросов к зачету

1. Понятия требований к разработке программного обеспечения
2. Классификация и уровни требований к разработке программного обеспечения.
3. Методологии и стандарты, регламентирующие работу с требованиями.
4. Принципы разработки программных приложений.
5. Методы организации работы в команде разработчиков ПО.
6. Системы контроля версий программного обеспечения.
7. Основные подходы к интегрированию программных модулей ПО.
8. Описание требований: унифицированный язык моделирования - краткий словарь.
9. Диаграммы UML.
10. Анализ требований и стратегии выбора решения.
11. Моделирование бизнес-процесса.
12. Диаграммы стандарта IDEF0
13. Виды, цели и уровни интеграции программных модулей.
14. Организация работы команды в системе контроля версий.
15. Инструментальные средства разработки программ. Классификация.
16. Необходимое и часто используемое ПО.
17. Специализированные интегрированные среды.
18. Универсальные характеристики инструментальных средств: фирма, производитель, автор; название продукта.
19. Универсальные характеристики инструментальных средств: класс продукта, который установил для него производитель; тип дистрибуции программы и стоимость;

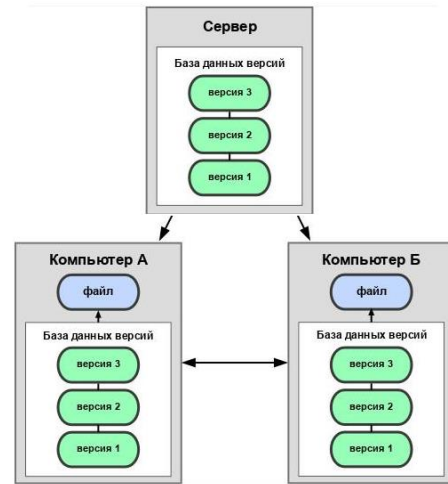
20. Универсальные характеристики инструментальных средств наличие и тип поддержки, ее стоимость; доступность и качество документации.
21. Универсальные характеристики инструментальных средств: простота и понятность интерфейса; наличие пробных версий (для платных программ); сайт программы и возможность ее скачивания; размер дистрибутива и его состав.
22. Интегрированная среда разработки ПО Microsoft Visual Studio. Структура Visual Studio. Инструменты среды разработки Visual Studio.

Перечень типовых заданий в тестовой форме

1. Принципы разработки программного обеспечения:
 - 1 Индивидуальный процесс разработки
 - 2 Командный процесс разработки
 - 3 Модель зрелости возможностей
 - 4 Модель законченности возможностей
2. Какие технологии разработки программ используются в современном программировании:
 - 1 Визуальные
 - 2 Структурные
 - 3 Событийные
 - 4 Объектно-ориентированные
3. Какие задачи решает система контроля версий:
 - 1 Защищает исходный код от потери.
 - 2 Обеспечивает командную работу.
 - 3 Помогает отменить изменения.
 - 4 Распределённая работа.
 - 5 Все ответы верные.
4. Установите соответствие между системами контроля версий

<p>Локальные системы контроля версий с простой базой данных, в которой хранятся все изменения нужных файлов</p>	
<p>Централизованные системы контроля версий. В этих системах есть центральный сервер, на котором хранятся все файлы под версионным контролем.</p>	

Распределённые системы контроля версий. В таких системах клиенты не просто выгружают последние версии файлов, а полностью копируют весь репозиторий.



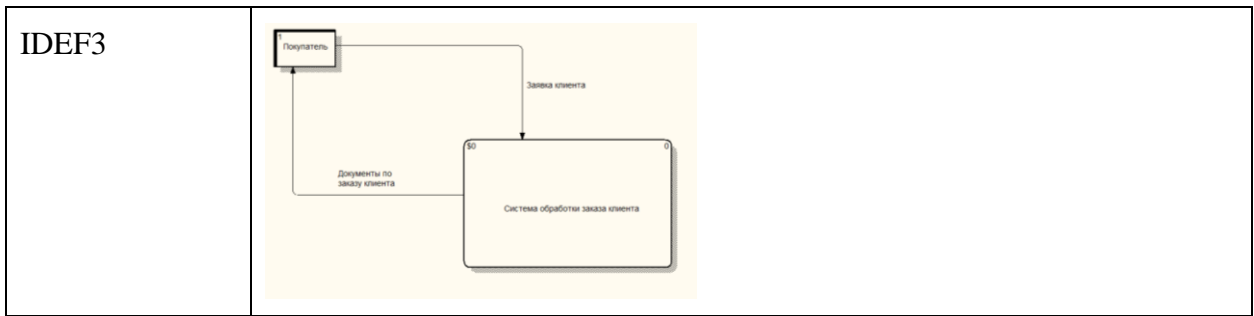
5. Установите соответствие между действием и базовой командой системы контроля версий Git

Клонировать репозиторий	git show -s
Обновить локальную сборку из репозитория	git commit -a -m "MESSAGE"
Проверить текущую версию (где я есть?)	git checkout BRANCH
Закоммитить изменения	git clone <откуда> <куда>
Переключиться на branch	git pull

Ответ: 1-4, 2-5, 3-1, 4-2, 5-3

6. Установите соответствие

IDEF1X	
IDEF0	
IDEF1	
DFD	



Ответ: 1-3,2-1,3-1,4-5,5-4

7. Разработка и сопровождение ИС в конкретной организации и конкретном проекте должна поддерживаться стандартами:

- 1 Стандарт организации
- 2 Стандарт проектирования
- 3 Стандарт оформления проектной документации
- 4 Стандарт пользовательского интерфейса

8. _____ – это необходимый процесс в ходе разработки, во время которого выявляются все проблемы в работе программного обеспечения.

Ответ: Тестирование

9. _____ — это процесс определения соответствия программного продукта потребностям пользователей, ее обычно проводят для конечного продукта в условиях, максимально приближенных к тем, в которых будет эксплуатироваться система.

Ответ: аттестация

10. _____ — это процесс подтверждения экспертизой того, что программным средством полностью реализуются предъявляемые к ней требования. Дополнительной целью является выявление и регистрация дефектов и ошибок которые были допущены во время разработки или модификации продукта.

Ответ: верификация

11. _____ — это программное обеспечение, которое обеспечивает командную работу в рамках одного или нескольких проектов.

Ответ: Система контроля версий

12. Требования к разрабатываемой системе должны включать:

- 1 Совокупность условий, при которых предполагается эксплуатировать будущую систему (аппаратные и программные ресурсы, предоставляемые системе; внешние условия ее функционирования; состав людей и работ, имеющих к ней отношение)
- 2 Описание выполняемых системой функций
- 3 Ограничения в процессе разработки (директивные сроки завершения отдельных этапов, имеющиеся ресурсы, организационные процедуры и мероприятия, обеспечивающие защиту информации)
- 4 Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения

13. К этапам развития технологии разработки программного обеспечения относятся:

- 1 «Процедурное» программирование
- 2 Программирование на алгоритмических языках высокого уровня
- 3 Структурный подход к программированию
- 4 Программирование на языках низкого уровня

5 Компонентный подход и CASE-технологии
- Машинно-ориентированное программирование

14. Структурный подход к программированию – это:
- 1 Совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения
 - 2 Подход, в основе которого лежит декомпозиция (разбиение на части) сложных систем с целью последующей реализации в виде отдельных небольших (до 40-50 операторов) подпрограмм
 - 3 Подход к решению задачи, требующий создание структурной схемы этапов работ по разработке программного обеспечения
 - 4 Подход, требующий представления задачи в виде иерархии подзадач простейшей структуры
- Ответ: 1,2,4
15. Объектный подход к программированию – это:
- 1 Технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств
 - 2 Технология создания сложного программного обеспечения, позволяющая вести практически независимую разработку отдельных частей (объектов) программы
 - 3 Технология создания сложного программного обеспечения, в основе которой лежат новые способы организации программ, основанные на механизмах наследования, полиморфизма, композиции, наполнения
 - 4 Технология создания сложного программного обеспечения, основанная на объектно-ориентированном программировании
16. Принципы разработки программного обеспечения:
- 1 Индивидуальный процесс разработки
 - 2 Командный процесс разработки
 - 3 Модель зрелости возможностей
 - 4 Модель законченности возможностей
17. Какие технологии разработки программ используются в современном программировании:
- 1 Визуальные
 - 2 Структурные
 - 3 Событийные
 - 4 Объектно-ориентированные
18. Что является результатом методологии IDEF0?
- 1 Функциональная модель;
 - 2 Реляционная таблица
 - 3 Модель IS- AS
 - 4 Модель IS- BE
19. Диаграмма IDEF0 может содержать следующие типы диаграмм:
- 1 Диаграмму классов
 - 2 Контекстную диаграмму, диаграмму декомпозиции
 - 3 Диаграмму компонентов
 - 4 Диаграмму дерева узлов

- 5 Диаграмму взаимодействий
- 6 Диаграмму только для экспозиции (FEO)
- 7 Диаграмму последовательности, диаграмму кооперации

20. Уровни логической модели:

- 1 Диаграмма сущность
- 2 Диаграмма связь
- 3 Диаграмма сущность-связь
- 4 Модель данных, основанная на ключах
- 5 Полная операционная модель
- 6 Полная атрибутивная модель

21. Модель DFD описывает:

- 1 Функции обработки стрелок (arrow)
- 2 Функции обработки информации (работы)
- 3 Внешние ссылки (external references), объекты, сотрудников или отделы, которые участвуют в обработке информации
- 4 Документы (стрелки, arrow), объекты, сотрудников или отделы, которые участвуют в обработке информации
- 5 Функции обработки внешних ссылок
- 6 Внешние ссылки (external references), таблицы для хранения документов (хранилище данных, data stor+ E)

Перечень типовых практических задач к дифференцированному зачету

Ситуационная задача № 1. Используя клиент командной строки Git или графический клиент (GitKraken, Sourcetree и т.д.), выполните следующие операции с локальным репозиторием:

1. Инициализируйте новый локальный репозиторий в пустой папке с именем my_project.
2. Создайте в этой папке текстовый файл с именем README.md.
3. Добавьте в файл README.md одну строку текста: # Мой первый проект.
4. Зафиксируйте текущее состояние файла в репозитории с комментарием «Добавлен файл README».
5. Внесите изменение в файл README.md, добавив вторую строку: ## Описание проекта.
6. Просмотрите статус репозитория и изменения в файлах с помощью соответствующей команды.
7. Зафиксируйте второе изменение с комментарием «Добавлено описание проекта».

Требования к результату:

1. Все операции должны быть выполнены с помощью команд Git.
2. Предоставьте преподавателю скриншот окна терминала (командной строки) с историей выполненных команд и результатами их выполнения (например, вывод команд git status и git log).
3. В истории коммитов (вывод команды git log --oneline) должно быть четко видно два коммита с указанными комментариями.

Ситуационная задача №2: Инициализация репозитория и выполнение базовых операций в системе контроля версий Git

Задание:

Используя клиент командной строки Git или графический клиент (GitKraken, Sourcetree и т.д.), выполните следующие операции с локальным репозиторием:

1. Инициализируйте новый локальный репозиторий в пустой папке с именем `my_project`.
2. Создайте в этой папке текстовый файл с именем `README.md`.
3. Добавьте в файл `README.md` одну строку текста: `# Мой первый проект`.
4. Зафиксируйте текущее состояние файла в репозитории с комментарием «Добавлен файл `README`».
5. Внесите изменение в файл `README.md`, добавив вторую строку: `## Описание проекта`.
6. Просмотрите статус репозитория и изменения в файлах с помощью соответствующей команды.
7. Зафиксируйте второе изменение с комментарием «Добавлено описание проекта».

Требования к результату:

1. Все операции должны быть выполнены с помощью команд `Git`.
2. Предоставьте преподавателю скриншот окна терминала (командной строки) с историей выполненных команд и результатами их выполнения (например, вывод команд `git status` и `git log`).
3. В истории коммитов (вывод команды `git log --oneline`) должно быть четко видно два коммита с указанными комментариями.

Ситуационная задача №3: Построение диаграммы последовательности (Sequence Diagram)

Задание:

Используя среду моделирования, постройте диаграмму последовательности для сценария «Оформление заказа в интернет-магазине» на основе следующего описания:

1. Пользователь добавляет товар в корзину.
2. Система проверяет наличие товара на складе.
3. Пользователь нажимает «Оформить заказ».
4. Система запрашивает у Платежного шлюза подтверждение оплаты.
5. Платежный шлюз возвращает статус «Оплата подтверждена».
6. Система создает новый заказ и отправляет уведомление на склад.

Требования к результату:

Диаграмма должна содержать объекты: `Пользователь`, `Интерфейс сайта`, `Система`, `Платежный шлюз`, `Склад`. Отобразите лимиты активации и сообщения между объектами в правильной последовательности. Сохраните диаграмму в формате PNG.

Ситуационная задача №4: Создание диаграммы классов (Class Diagram) по описанию

Задание:

На основе текстового описания постройте диаграмму классов в UML.

- В системе есть класс `Book` (Книга) с атрибутами: `title` (строка), `author` (строка), `isbn` (строка).
- Класс `Library` (Библиотека) содержит коллекцию (массив) объектов `Book`.
- Класс `Library` имеет методы: `addBook(book: Book)`, `findBookByTitle(title: String): Book`.
- Между `Library` и `Book` существует отношение композиции (целое-часть).

Требования к результату:

Отобразите классы с их атрибутами и методами, типы данных и `visibility` (`-private`, `+public`). Правильно изобразите связь композиции. Сохраните работу в исходном файле программы.

Ситуационная задача №5: Построение диаграммы развертывания (Deployment Diagram)

Задание:

Смоделируйте диаграмму развертывания для простого веб-приложения.

- Приложение состоит из Web-сервера (Apache), Сервера приложений (Node.js) и Базы данных (MySQL).
- Клиентские рабочие станции взаимодействуют с Web-сервером через протокол HTTP.
- Web-сервер взаимодействует с Сервером приложений через внутреннюю сеть.
- Сервер приложений подключается к Базе данных по специализированному соединению.

Требования к результату:

Используйте стандартные UML-артефакты (` < `) и связи между ними с указанием протоколов. Сохраните диаграмму в формате JPEG.

Ситуационная задача №6: Клонирование репозитория и создание новой ветки в Git

Задание:

1. Используя командную строку Git, клонируйте репозиторий по предоставленной преподавателем ссылке.
2. Перейдите в директорию скопированного проекта.
3. Создайте новую ветку с именем feature/add-new-button.
4. Сделайте коммит в новой ветке с любым изменением (например, создайте текстовый файл change.log).
5. Отправьте (push) созданную ветку на удаленный сервер.

Требования к результату:

Предоставьте скриншот терминала с последовательностью выполненных команд и их выводом (команды git clone, git branch, git checkout, git push).

Ситуационная задача №7: Построение контекстной диаграммы IDEF0

Задание:

Для системы «Онлайн-банкинг» постройте контекстную диаграмму IDEF0 (уровень A-0). Главная функция (функция-цель) – «Обеспечивать финансовые операции клиента».

Требования к результату:

Диаграмма должна содержать:

- **Входы:** Запрос на операцию, данные учетной записи.
 - **Выходы:** Результат операции, выписка, уведомление.
 - **Управление:** Правила банка, финансовое законодательство.
 - **Механизмы:** Клиент, банковский сервер.
- Укажите стандартные стрелки и их назначение. Сохраните в PDF.

Ситуационная задача №8: Построение диаграммы потоков данных (DFD уровня 0)

Задание:

На основе контекстной диаграммы IDEF0 из Задачи №7 постройте диаграмму потоков данных (DFD) уровня 0 для той же системы «Онлайн-банкинг».

Требования к результату:

Диаграмма должна отображать основные процессы (например, «Аутентификация пользователя», «Обработка перевода», «Формирование отчетности»), внешние сущности («Клиент», «ЦБ РФ») и потоки данных между ними. Используйте стандартные обозначения Gane-Sarson или Yourdon. Сохраните в PNG.

Ситуационная задача №9: Создание и слияние веток в Git (Merge)

Задание:

1. В существующем локальном репозитории создайте две новые ветки: `bugfix/login-error` и `hotfix/security-patch`.
2. В каждой ветке создайте по одному уникальному файлу (например, `fix_login.txt` и `patch_security.txt`) и сделайте по коммиту.
3. Переключитесь на ветку `main` (или `master`).
4. Выполните слияние (`merge`) ветки `hotfix/security-patch` с веткой `main`.
5. Разрешите `hypothetical merge conflict` (если смоделирован) и сделайте коммит слияния.

Требования к результату:

Предоставьте скриншот истории коммитов (команда `git log --oneline --graph --all`), наглядно показывающий созданные ветки и процесс их слияния.

Ситуационная задача №10: Построение диаграммы состояний (Statechart Diagram)

Задание:

Постройте диаграмму состояний для объекта «Заказ» в интернет-магазине.

- Начальное состояние: Создан.
- Возможные состояния: Оплачен, Передан в доставку, Доставлен, Отменен.
- События: Оплата получена, Отправлен курьеру, Получен клиентом, Покупатель отказался.

Требования к результату:

Диаграмма должна отображать все состояния, переходы между ними по событиям и финальное состояние. Сохраните в формате исходного файла программы (например, `.vpp`).

Ситуационная задача №11: Построение диаграммы компонентов (Component Diagram)

Задание:

Смоделируйте диаграмму компонентов для приложения «Погодный виджет».

- Компоненты: `UIInterface`, `WeatherDataParser`, `NetworkClient`, `DataStorage`.
- Покажите зависимости между компонентами (например, `UIInterface` зависит от `WeatherDataParser`).
- Укажите предоставляемые и требуемые интерфейсы для компонента `WeatherDataParser`.

Требования к результату:

Используйте стандартные обозначения компонентов, интерфейсов (`lollipop`) и зависимостей. Подпишите все элементы. Сохраните в JPEG.

Ситуационная задача №12: Форматирование и коммит изменений с помощью .gitignore

Задание:

1. В существующем Git-репозитории создайте файл `.gitignore`.
2. Настройте его так, чтобы Git игнорировал все файлы с расширением `.log` и содержимое папки `temp/`.
3. Создайте несколько тестовых файлов (например, `debug.log`, `temp/test.txt`) и убедитесь, что они не отслеживаются (имеют статус `untracked` в `git status`).
4. Добавьте и закомитуйте только файл `.gitignore`.

Требования к результату:

Предоставьте скриншот вывода команды `git status`, который показывает, что файлы `.log` и в папке `temp/` не добавлены к отслеживанию, а файл `.gitignore` – готов к коммиту.

Критерии и шкала оценки дифференцированного зачета

Оценка	Характеристики ответа обучающегося
Отлично	- студент глубоко и всесторонне усвоил программный материал;

	<ul style="list-style-type: none"> - уверенно, логично, последовательно и грамотно его излагает; - опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью; - умело обосновывает и аргументирует выдвигаемые им идеи; - делает выводы и обобщения; - свободно владеет системой понятий по дисциплине; - правильно решил ситуационную задачу и прошел тестирование.
Хорошо	<ul style="list-style-type: none"> - студент твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; - не допускает существенных неточностей; - увязывает усвоенные знания с практической деятельностью; - аргументирует научные положения; - делает выводы и обобщения; - владеет системой понятий по дисциплине; - правильно решил ситуационную задачу и прошел тестирование.
Удовлетворительно	<ul style="list-style-type: none"> - студент усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; - допускает несущественные ошибки и неточности; - испытывает затруднения в практическом применении знаний; - слабо аргументирует научные положения; - затрудняется в формулировании выводов и обобщений; - частично владеет системой понятий по дисциплине; - с затруднениями решил ситуационную задачу, ошибся в тестировании..
Неудовлетворительно	<ul style="list-style-type: none"> - студент не усвоил значительной части программного материала; - допускает существенные ошибки и неточности при рассмотрении проблем; - испытывает трудности в практическом применении знаний; - не может аргументировать научные положения; - не формулирует выводов и обобщений; - не решил ситуационную задачу, не прошел тестирование.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Основная литература

1. Грекул, В. И. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 404 с. — (Профессиональное образование). — ISBN 978-5-534-19506-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/566739>

2. 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, 2024. — 400 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0812-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2136716>

3. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 273 с. — (Профессиональное образование). — ISBN 978-5-534-20362-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562355>

8.2.Дополнительная литература

1. Черткова, Е. А. Программная инженерия. Визуальное моделирование программных систем : учебник для среднего профессионального образования / Е. А. Черткова. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 146 с. — (Профессиональное образование). — ISBN 978-5-534-18094-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563828>

2. Гниденко, И. Г. Технология разработки программного обеспечения : учебник для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 248 с. — (Профессиональное образование). — ISBN 978-5-534-18131-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563151>



Периодические издания:

– Прикладная информатика : научно-информационный журнал / Издательство университет «Синергия». – 2006. – Москва, 2006-2025. – ISSN 1993-8314. - Текст : электронный. - URL: <http://www.iprbookshop.ru/11770.html>

– IT Expert : журнал «Экспресс Электроника» / Издательство ИТ Медиа. - 1993. - Санкт-Петербург, 2009-2022. - Текст электронный. URL: <https://www.iprbookshop.ru/38869.html>

8.3. Программное обеспечение

- Microsoft Windows,
- Microsoft Office или Яндекс 360;
- Microsoft Visio;
- Draw.io Desktop.

8.4. Базы данных, информационно-справочные и поисковые системы, Интернет-ресурсы

Базы данных (профессиональные базы данных)

–База данных IT специалиста– Режим доступа: <http://info-comp.ru/>

Информационно-справочные системы

–Электронная библиотечная система «СКСИ» - <https://www.sksi.ru/Environment/EbsSksi>

–Информационно-справочная система для программистов - <http://life-prog.ru>

Поисковые системы

- Поисковая система Google - <https://www.google.ru>

- Поисковая система Yandex - <https://www.yandex.ru>
- Поисковая система Rambler - <http://www.rambler.ru>
- Электронные образовательные ресурсы*
- Научная электронная библиотека «Киберленинка» – <http://cyberleninka.ru/>
- Национальный открытый университет Интуит – <http://www.intuit.ru/>
- Цифровой образовательный ресурс IPR SMART – <https://www.iprbookshop.ru/>
- Образовательная платформа Юрайт - <https://urait.ru/>
- Электронно-библиотечная система Znanium - <https://znanium.com/>

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины необходимо следующее материально-техническое обеспечение:

- для проведения лекций, уроков – аудитория, укомплектованная оборудованием и техническими средствами обучения: учебная мебель, экран, проектор, компьютер, расходный материал;

- для проведения всех видов практических занятий – компьютерный класс с лицензионным программным обеспечением, укомплектованный оборудованием и техническими средствами обучения: учебная мебель, экран, проектор, компьютеры (с лицензионным программным обеспечением), расходный материал;

- для текущего контроля и промежуточной аттестации – компьютерный класс с лицензионным программным обеспечением, укомплектованный оборудованием и техническими средствами обучения: учебная мебель, экран, проектор, компьютеры (с лицензионным программным обеспечением), расходный материал;

- для проведения практической подготовки – компьютерный класс с лицензионным программным обеспечением, укомплектованный оборудованием и техническими средствами обучения: учебная мебель, экран, проектор, компьютеры (с лицензионным программным обеспечением), расходный материал;

- для проведения индивидуальных и групповых консультаций – компьютерный класс с лицензионным программным обеспечением, укомплектованный оборудованием и техническими средствами обучения: учебная мебель, экран, проектор, компьютеры (с лицензионным программным обеспечением), расходный материал;

- для организации самостоятельной работы – помещение, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Института.

10. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ЛИЦАМИ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Обучающимся с ограниченными возможностями здоровья предоставляются специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, услуги ассистента (тьютора), оказывающего обучающимся необходимую техническую помощь, а также услуги сурдопереводчиков и тифлосурдопереводчиков.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья и инвалидами может быть организовано совместно с другими обучающимися, а также в отдельных группах.

Освоение дисциплины обучающимися с ограниченными возможностями здоровья и инвалидами осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья.

В целях доступности получения среднего профессионального образования по образовательной программе лицами с ограниченными возможностями здоровья при освоении дисциплины обеспечивается:

1) для лиц с ограниченными возможностями здоровья по зрению:

– присутствие тьютора, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– письменные задания, а также инструкции о порядке их выполнения оформляются увеличенным шрифтом,

– специальные учебники, учебные пособия и дидактические материалы (имеющие крупный шрифт или аудиофайлы),

– индивидуальное равномерное освещение не менее 300 люкс,

– при необходимости студенту для выполнения задания предоставляется увеличивающее устройство;

2) для лиц с ограниченными возможностями здоровья по слуху:

– присутствие ассистента, оказывающий студенту необходимую техническую помощь с учетом индивидуальных особенностей (помогает занять рабочее место, передвигаться, прочитать и оформить задание, в том числе, записывая под диктовку),

– обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающемуся предоставляется звукоусиливающая аппаратура индивидуального пользования;

– обеспечивается надлежащими звуковыми средствами воспроизведения информации;

3) для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата:

– письменные задания выполняются на компьютере со специализированным программным обеспечением или надиктовываются тьютору;

– по желанию студента задания могут выполняться в устной форме.

Программа составлена в соответствии с требованиями ФГОС СПО по специальности 09.02.11 Разработка и управление программным обеспечением и ПОП.