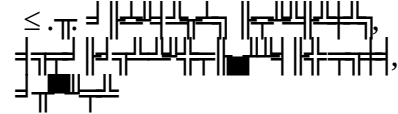


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Порохня Андрей Алексеевич
Должность: И.о. директора института перспективной инженерии
Дата подписания: 02.06.2026 19:53:32
Уникальный программный ключ:
990bea3aeec848c23bd8699e48288f8d1960c608

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ


А.А. Порохня
Ф.И.О.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
Технологии программирования
название дисциплины

Направление подготовки	09.04.03 Прикладная информатика
Направленность (профиль)	Управление знаниями
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	3

Введение

1. Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

2. ФОС является приложением к программе дисциплины «Технологии программирования»

3. Разработчик Амироков С.Р., доцент.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель Азаров И.В., и.о. директора департамента

Члены комиссии:

Орлова А.Ю., доцент

Орлинская О.Г., доцент

Представитель организации-работодателя Черевко С.А., генеральный директор ООО "КРАФТКОД"

(Ф.И.О., должность)

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

5. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Уровни сформированности и компетенции(ий), индикатора (ов)	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ОПК-2				
ИД-1 ОПК-2. разрабатывать оригинальные алгоритмы для решения профессиональных задач;	С трудом разрабатывает оригинальные алгоритмы для решения профессиональных задач	Не в полном объеме разрабатывает оригинальные алгоритмы для решения профессиональных задач	В целом разрабатывает оригинальные алгоритмы для решения профессиональных задач	разрабатывает оригинальные алгоритмы для решения профессиональных задач
ИД-2 ОПК-2. разрабатывать программные средства для решения профессиональных задач	С трудом разрабатывает программные средства для решения профессиональных задач	Слабо разрабатывает программные средства для решения профессиональных задач	В целом разрабатывает программные средства для решения профессиональных задач	разрабатывает программные средства для решения профессиональных задач
ИД-3 ОПК-2. использовать современные интеллектуальные технологии для решения профессиональных задач	С трудом использует современные интеллектуальные технологии для решения профессиональных задач	Слабо использует современные интеллектуальные технологии для решения профессиональных задач	В целом использует современные интеллектуальные технологии для решения профессиональных задач	использует современные интеллектуальные технологии для решения профессиональных задач
ОПК-5				
ИД-1 ОПК-5. разрабатывать программное и аппаратное обеспечение информационных и автоматизированных систем.	С трудом разрабатывает программное и аппаратное обеспечение информационных и автоматизированных систем.	Слабо разрабатывает программное и аппаратное обеспечение информационных и автоматизированных систем.	В основном разрабатывает программное и аппаратное обеспечение информационных и автоматизированных систем.	разрабатывает программное и аппаратное обеспечение информационных и автоматизированных систем.
ИД-2 ОПК-5. модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	С трудом модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем	Слабо модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем	В основном модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем	модернизирует программное и аппаратное обеспечение информационных и автоматизированных систем
ПК-7				
ИД-1 ПК-7. организовывать переговоры с представителями заказчика	Не может организовать переговоры с представителями заказчика	Допускает серьезные ошибки в организации переговоров с представителями заказчика	В основном организует переговоры с представителями заказчика	организует переговоры с представителями заказчика

ИД-2 ПК-7. проводить переговоры с представителями заказчика	С трудом проводить переговоры с представителями заказчика	Допускает серьезные ошибки в проведении переговоров с представителями заказчика	В основном проводить переговоры с представителями заказчика	проводить переговоры с представителями заказчика
ИД-3 ПК-7. Организовывать профессиональные консультации на предприятиях и в организациях	С трудом организует профессиональные консультации на предприятиях и в организациях	Допускает серьезные ошибки в организации профессиональных консультаций на предприятиях и в организациях	В основном организует профессиональные консультации на предприятиях и в организациях	организует профессиональные консультации на предприятиях и в организациях
ПК-9				
ИД-1 ПК-9. разрабатывать информационные системы, их компоненты и информационные сервисы	С трудом разрабатывает информационные системы, их компоненты и информационные сервисы	Слабо разрабатывает информационные системы, их компоненты и информационные сервисы	В целом разрабатывает информационные системы, их компоненты и информационные сервисы	разрабатывает информационные системы, их компоненты и информационные сервисы
ИД-2 ПК-9. интегрировать информационные системы, их компоненты и информационные сервисы	С трудом интегрирует информационные системы, их компоненты и информационные сервисы	Слабо интегрирует информационные системы, их компоненты и информационные сервисы	В целом интегрирует информационные системы, их компоненты и информационные сервисы	интегрирует информационные системы, их компоненты и информационные сервисы
ПК-10				
ИД-1 ПК-10. развертывать, информационные системы, их компоненты и информационные сервисы	С трудом развертывает, информационные системы, их компоненты и информационные сервисы	Слабо развертывает, информационные системы, их компоненты и информационные сервисы	В целом развертывает, информационные системы, их компоненты и информационные сервисы	развертывает, информационные системы, их компоненты и информационные сервисы
ИД-2 ПК-10 инсталлировать информационные системы, их компоненты и информационные сервисы	С трудом инсталлирует информационные системы, их компоненты и информационные сервисы	Слабо инсталлирует информационные системы, их компоненты и информационные сервисы	В целом инсталлирует информационные системы, их компоненты и информационные сервисы	инсталлирует информационные системы, их компоненты и информационные сервисы
ПК-11				
ИД-1 ПК-11. гарантировать качество ИС	С трудом гарантирует качество ИС	Слабо гарантирует качество ИС	В целом гарантирует качество ИС	гарантирует качество ИС
ИД-2 ПК-11. гарантировать надежность ИС	С трудом гарантирует надежность ИС	Слабо гарантирует надежность ИС	В целом гарантирует надежность ИС	гарантирует надежность ИС.
ИД-3 ПК-11. гарантировать информационную безопасность ИС	С трудом гарантирует информационную безопасность ИС	Слабо гарантирует информационную безопасность ИС	В целом гарантирует информационную безопасность ИС	гарантирует информационную безопасность ИС

Оценивание уровня сформированности компетенции по дисциплине осуществляется на основе «Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры - в федеральном государственном автономном образовательном учреждении высшего образования «Северо-Кавказский федеральный университет» в актуальной редакции.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Но ме р зад ан ия	Правильный ответ	Содержание вопроса	Компетенция
		Форма обучения очная, семестр 3	
1.	1	Ограничение количества команд при диалоге на основе командного языка проводится для того, чтобы не (1) Перегружать память пользователя (2) Зависеть от возможностей операционной системы (3) Предъявлять высоких требований к программным ресурсам (4) Занимать оперативную память компьютера	ОПК-2
2.	1	В объектно-ориентированных языках операции над объектом являются составной частью: (1) Класса (2) Метода (3) Объекта	ПК-9
3.	2	На аналогии с обычным интервью основана структура диалога (1) На основе экранных форм (2) Типа «вопрос-ответ» (3) На основе командного языка (4) Типа «задача-решение»	ОПК-2
4.	1	Язык Пролог предназначен для решения задач (1) Логических (2) Физических (3) Концептуальных	ОПК-2
5.	1,4	При трансляции описания программы, построенной с помощью 4GL, создаются (1) Исходный текст программы (2) Объектный модуль (3) Загрузочный модуль	ОПК-5

		(4) Файл описания ресурсов интерфейса	
6.	2	Разработка и отладка отдельной функции могут проводиться после разработки главной программы (1) Верно (2) Неверно	ОПК-2
7.	2	Начиная с головного модуля с последовательным подключением других модулей проводится стратегия тестирования (1) Исходящего (2) Нисходящего (3) Восходящему (4) Комбинированному	ОПК-5
8.	1	Совокупность записей данных, передаваемых другому приложению или получаемых от него, называется (1) Интерфейс приложения (2) Логический файл (3) Выходной элемент приложения (4) Запрос	ПК-9
9.	2	В объектно-ориентированных языках операции над объектом называются: (1) Классами (2) Методами (3) Объектами	ОПК-2
10.	2	В случае отсутствия исходного кода для соответствующего модуля используется точка останова: (1) По классу (2) По адресу (3) По объекту	ОПК-2
11.	3	Абстракция через спецификацию заключается в том, что (1) Одним алгоритмом можно решать задачи с разными исходными данными (2) Алгоритм может работать только при определенном диапазоне исходных данных (3) Разными алгоритмами можно получить один и тот же результат	ПК-9

12.	4	При отладке вызовы функций и процедур, выполненные до определённого момента, можно посмотреть с помощью (1) Стека вызовов (2) Журнала событий (3) Просмотра потоков (4) Окона CPU	ОПК-5
13.	4	Количество этапов в процессе тестирования (1) 5 (2) 2 (3) 4 (4) 3	ОПК-5
14.	1	Методологии определяют языки программирования и системы (1) Да (2) Нет	ОПК-2
15.	1	Общедоступными являются все компоненты класса (1) Верно (2) Неверно	ОПК-2
16.	1	Тестирование по маршрутам, предусмотренным алгоритмом, применяют при тестировании (1) Структурном (2) Функциональном (3) Комплексном (4) Оценочным	ОПК-5
17.	1	Основное значение логического программирования - разработка интеллектуальных систем (1) Да (2) Нет	ОПК-2
18.	4	Метод формирования тестовых наборов, при котором в наборы включают один правильный класс и все возможные неправильные, – это (1) Анализ граничных значений (2) Покрытие решений (3) Покрытие условий (4) Эквивалентное разбиение	ОПК-5

19.	4	После исправления ошибки тестирование (1) Продолжают по плану без повторения предыдущих тестов (2) Проводят на данных, на которых выявлена ошибка, и продолжают далее по плану (3) Проводят только на данных, для которых выявлена ошибка (4) Повторяют сначала	ОПК-5
20.	2	Допустимое время ответа интерактивной системы для подтверждения физических действий составляет (1) 1-2 сек (2) 0,1-0,2 сек (3) 0,2-0,5 сек (4) 0,5-1,0 сек	ПК-10
21.	1	Тестирование - процесс исполнения программы в целях обнаружения ошибок (1) Верно (2) Неверно	ПК-9
22.	4	Каждый переключатель снабжается (1) Флажком (2) Полосой прокрутки (3) Шкалой (4) Текстовой меткой	ПК-10
23.	2	При использовании отладчика можно остановить выполнение программы (1) По достижении строки кода при выполнении условия (2) Перед выполнением определенной строки кода (3) В любой момент при щелчке мыши (4) В любой момент при нажатии определенной клавиши	ПК-7
24.	4	К возможностям, обеспечиваемым средствами отладки современных сред программирования, относятся (1) Выполнение программы по шагам (2) Пропуск выполнения фрагментов программы (3) Блокирования выполнения программы (4) Модифицирование значений переменных	ПК-9
25.	2	Ошибки компоновки обнаруживаются	ПК-9

		(1) Операционной системой (2) Компоновщиком (3) Пользователем при выполнении программы (4) Компилятором	
26.	2	Вторичное окно может быть модальным по отношению к: 1) своему первичному окну; 2) любому первичному окну; 3) любому вторичному окну; 4) системе - из перечисленного (1) 2,4 (2) 1,4 (3) 1,3 (4) 2,3	ПК-10
27.	2	Надёжность характеризуется (1) Устойчивостью к дефектам (2) Восстанавливаемостью (3) Стабильностью (4) Использованию ресурсов	ПК-9
28.	1	Ошибки компоновки связаны с (1) Разрешением внешних ссылок (2) Нечетким знанием правил языка программирования (3) Попыткой записи на устройства, защищённые от записи (4) Нарушением адресации	ОПК-2
29.	3	Типами структуры диалога являются: 1) на основе экранных форм; 2) на основе графического интерфейса; 3) на основе меню; 4) «вопрос-ответ»; 5) «задача-решение»; 6) на основе командного языка - из перечисленного (1) 1,3,4,6 (2) 2,4,5,6 (3) 1,2,3,5 (4) 1,3,5,6	ПК-10
30.	1	Структурное программирование начало развиваться (1) В 60-е годы XX века (2) В 80-е годы XX века (3) В 70-е годы XX века (4) В 50-е годы XX века	ОПК-2

31.	1	Стадии состоят из этапов, которые обычно имеют итерационный характер (1) Да (2) нет	ПК-9
32.	2	Выделяют две принципиально различающихся группы характеристик качества (1) Категорийные (2) Конструктивные (3) Функциональные (4) Количественные	ПК-7
33.	4	Свойство согласованности означает, что (1) Части программы взаимосвязаны (2) В программе нет ничего лишнего (3) В программе есть все что нужно (4) Программа и ее части соответствуют правилам и законам внешнего мира	ПК-11
34.	1	Проблемно-ориентированная помощь представляет собой описание последовательности шагов, необходимых для выполнения задания пользователя (1) Да, проблемно-ориентированная помощь представляет собой описание последовательности шагов, необходимых для выполнения задания пользователя (2) Нет, проблемно-ориентированная помощь не представляет собой описание последовательности шагов, необходимых для выполнения задания пользователя	ОПК-2
35.	1	Качество проекта программного продукта, от которого зависят трудовые и материальные затраты на его реализацию и последующие модификации, называют (1) Технологичностью (2) Согласованностью (3) Проработанностью (4) Совместимостью	ПК-11
36.	1	К стратегиям тестирования не относится программирование (1) Локализованное	ОПК-5

		(2) Детерминированное (3) Ручное (4) Стохастическое	
37.	3	К ошибкам выполнения всегда относятся (1) Логические ошибки (2) Неправильное применение конструкций языка программирования (3) Неверное определение исходных данных (4) Отсутствие подключаемого модуля	ПК-11
38.	1	Динамические библиотеки подключаются к программе в процессе (1) Выполнения программы (2) Компиляции (3) Компоновки (4) Создания проекта	ОПК-2
39.	2	Ошибки межмодульного интерфейса относятся к ошибкам (1) Синтаксическим (2) Выполнения (3) Семантическим (4) Компоновки	ОПК-2
40.	2	Модули динамически подключаемых библиотек создаются (1) Компилятором (2) Компоновщиком (3) Интерпретатором (4) Загрузчиком	ОПК-2
41.	2	Имя класса - произвольно выбираемый идентификатор (1) Неверно (2) Верно	ОПК-2
42.	1	К типу документации, имеющей оригинальный состав и не стандартизованной, относится (1) Исследовательская (2) Технологическая (3) Пользовательская (4) Эксплуатационная	ПК-7

43.	3	Описание алгоритмов с обоснованием принятых решений содержится в документе (1) Руководство программиста (2) Описание языка (3) Пояснительная записка (4) Описание применения	ПК-7
44.	1,4	К методам ручного контроля, при которых группа специалистов просматривает текст программы в целях обнаружения ошибок, относятся (1) Инспекции исходного текста (2) Проверка за столом (3) Оценка программ (4) Сквозные просмотры	ПК-10
45.	1	Метод «чёрного ящика» используется при подходе к формированию тестовых наборов данных (1) Функциональном (2) Объектно-ориентированном (3) Структурном (4) Сеансовом	ОПК-5
46.	1	Пользователь явно выбирает уровень диалоговой поддержки при адаптации (1) Фиксированной (2) Формальной (3) Полной косметической	ПК-7
47.	2	Целями разработки сценария диалога являются: 1) выявление и устранение возможных тупиковых ситуаций; 2) выбор рациональных путей перехода из одного состояния диалога в другое; 3) оптимизация внешнего вида интерфейса; 4) выявление неоднозначных ситуаций; 5) графическая интерпретация действий пользователя - из перечисленного (1) 2,3,5 (2) 1,2,4 (3) 1,3,4 (4) 2,3,4,5	ПК-11

48.	1	В заголовке окна сообщения нельзя использовать слово (1) Ошибка (2) Отмена (3) Введите (4) Сообщение	ОПК-2
49.	1	При использовании виртуальных функций сообщения передаются напрямую объекту (1) Неверно (2) Верно	ОПК-2
50.	3	Главной целью применения систем программирования в большинстве случаев является (1) Создание комфортных условий для программиста (2) Эффективность результирующей программы (3) Снижение трудозатрат при создании программ (4) Экономия ресурсов компьютера, используемых при разработке	ПК-11
51.	логической группировки нескольких файлов, содержащих исходный код, на одном из поддерживаемых языков программирования, а также любых вспомогательных файлов (ресурсы для проекта: это могут быть и дополнительные библиотеки, файлы изображений, иконок и прочее). Обычно после сборки проекта (которая включает компиляцию всех входящих в проект файлов исходного кода) создается один исполняемый модуль (исполняемый файл), либо несколько, если это веб-приложение.	Проект (Project) используется в Visual Studio для	ОПК-2
52.	это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования	Объектно-ориентированное программирование ООП	ПК-9

53.	это нечто, имеющее четко определенные границы. Однако, этого недостаточно, чтобы отделить один объект от другого или дать оценку качества абстракции. Объект обладает состоянием, поведением и идентичностью; структура и поведение схожих объектов определяет общий для них класс; термины «экземпляр класса» и «объект» взаимозаменяемы	Понятие объекта в ООП	ОПК-2
54.	это множество объектов, обладающих общей структурой, поведением и семантикой. Отдельный объект – это экземпляр класса. Класс представляет лишь абстракцию существенных свойств объекта	Понятие класса в ООП	ОПК-2
55.	представляет собой способность производить новый класс из существующего базового класса. Производный класс – это новый класс, а базовый класс – существующий класс. Когда вы порождаете один класс из другого (базового класса), производный класс наследует элементы базового класса. Для порождения класса из базового начинайте определение производного класса ключевым словом class, за которым следует имя класса, двоеточие и имя базового класса, например <code>class dalmatian: dog</code>	Наследование	ОПК-5
56.	выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя.	Понятие абстракции	ОПК-2

	Абстракция основывается на понятиях клиента и сервера. Клиентом называется любой объект, использующий ресурсы другого объекта (называемого сервером).		
57.	это процесс отделения друг от друга элементов объекта, определяющих его устройство и поведение. Инкапсуляция служит для того, чтобы изолировать контрактные обязательства абстракции от их реализации. Абстракция и инкапсуляция дополняют друг друга: абстрагирование направлено на наблюдаемое поведение объекта, а инкапсуляция занимается внутренним устройством. Чаще всего инкапсуляция выполняется посредством скрытия информации, то есть маскировкой всех внутренних деталей, не влияющих на внешнее поведение. Обычно скрываются и внутренняя структура объекта, и реализация его методов. Практически это означает наличие двух частей в классе: интерфейса и реализации. Интерфейс отражает внешнее поведение объекта, описывая абстракцию поведения всех объектов данного класса. Внутренняя реализация описывает представление этой абстракции и механизмы достижения желаемого поведения объекта. Принцип разделения интерфейса и реализации соответствует сути вещей: в интерфейсной части собрано все, что касается взаимодействия данного объекта с любыми другими объектами; реализация скрывает от других объектов все детали, не имеющие	Понятие инкапсуляции	ОПК-5

	отношения к процессу взаимодействия объектов.		
58.	это упорядочение абстракций, расположение их по уровням. Основными видами иерархических структур применительно к сложным системам являются структура классов (иерархия "is-a") и структура объектов (иерархия "part of"). Важным элементом объектно-ориентированных систем и основным видом иерархии "is-a" является упоминавшаяся выше концепция наследования. Наследование означает такое отношение между классами (отношение родитель/потомок), когда один класс заимствует структурную или функциональную часть одного или нескольких других классов (соответственно, одиночное и множественное наследование). Иными словами, наследование создает такую иерархию абстракций, в которой подклассы наследуют строение от одного или нескольких суперклассов. Часто подкласс достраивает или переписывает компоненты вышестоящего класса. Если иерархия "is a" определяет отношение "обобщение/специализация", то отношение "part of" (часть) вводит иерархию агрегации. В иерархии "part of" класс находится на более высоком уровне абстракции, чем любой из использовавшихся при его реализации.	Понятие иерархии	ПК-9

59.	- есть отражение фон Неймановской архитектуры компьютера. Программа, написанная на процедурном языке, представляет собой последовательность команд, определяющих алгоритм решения задачи. Основная команда присвоение, с помощью которой определяется и меняется память компьютера. Программа производит преобразование содержимого памяти, изменяя его от исходного состояния к результирующему.	Процедурное программирование	ОПК-2
60.	— это функция или процедура, принадлежащая какому-то классу или объекту. Как и процедура в процедурном программировании, метод состоит из некоторого количества операторов для выполнения какого-то действия и имеет набор входных аргументов.	Метод в объектно-ориентированном программировании	ОПК-2
61.	это свойство, которое позволяет одно и то же имя использовать для решения нескольких технически разных задач. В общем смысле, концепцией полиморфизма является идея "один интерфейс, множество методов". Это означает, что можно создать общий интерфейс для группы близких по смыслу действий. Преимуществом полиморфизма является то, что он помогает снижать сложность программ, разрешая использование одного интерфейса для единого класса действий. Выбор конкретного действия, в зависимости от ситуации, возлагается на компилятор.	Понятие полиморфизма	ПК-9

62.	• уменьшение сложности программного обеспечения; • повышение надежности программного обеспечения; • обеспечение возможности модификации отдельных компонентов программного обеспечения без изменения остальных его компонентов; • обеспечение возможности повторного использования отдельных компонентов программного обеспечения.	Преимущества объектно-ориентированного подхода	ОПК-5
63.	Весь период существования ПО, связанный с подготовкой к его разработке, разработкой, использованием и модификациями, начиная с того момента, когда принимается решение разработать/приобрести/собрать из имеющихся компонентов новую систему или приходит сама идея о необходимости программы определенного рода, до того момента, когда полностью прекращается всякое ее использование	Жизненный цикл программного обеспечения	ОПК-5
64.	создаваемые человеком информационные сущности, документы в достаточно общем смысле, участвующие в качестве входных данных и получающиеся в качестве результатов различных деятельности. Примерами артефактов являются: модель предметной области, описание требований, техническое задание, архитектура системы, проектная документация на систему в целом и на ее компоненты, прототипы системы и компонентов, собственно, исходный код,	Понятие артефакта в программном обеспечении	ОПК-2

	пользовательская документация, документация администратора системы, руководство по развертыванию, база пользовательских запросов, план проекта, и пр.		
65.	Верификация обозначает проверку того, что ПО разработано в соответствии со всеми требованиями к нему, или что результаты очередного этапа разработки соответствуют ограничениям, сформулированным на предшествующих этапах	Понятие процесса верификации	ОПК-2
66.	Валидация — это проверка того, что сам продукт правилен, т.е. подтверждение того, что он действительно удовлетворяет потребностям и ожиданиям пользователей, заказчиков и других заинтересованных сторон	Понятие процесса валидации	ОПК-5
67.	(class diagrams) показывают классы или типы сущностей системы характеристики классов (поля и операции) и возможные связи между ними. Классы представляются прямоугольниками, поделенными на три части. В верхней части показывают имя класса, в средней — набор его полей, с именами, типами, модификаторами доступа (public '+', protected '#', private '-') и начальными значениями, в нижней — набор операций класса. Для каждой операции показывается ее модификатор доступа и сигнатура.	Диаграммы классов	ОПК-2
68.	показывают часть объектов системы и с ними в некотором конкретном состоянии или суммарно, за некоторый интервал времени. Объекты изображаются	Диаграммы объектов	ОПК-5

	прямоугольниками с идентификаторами ролей объектов (в контексте тех состояний, которые изображены на диаграмме) и типами. Однородные коллекции объектов могут изображаться накладывающимися друг на друга прямоугольниками.		
69.	представляют компоненты в нескольких — атомарные составляющие системы с точки зрения ее сборки, конфигурационного управления и развертывания. Компоненты сборки и конфигурационного управления обычно представляют собой файлы с исходным кодом, динамически подгружаем пр., компоненты библиотеки, HTML-странички и развертывания — это компоненты JavaBeans, CORBA, COM и т.д.	Диаграммы компонентов	ОПК-5
70.	показывают декомпозицию системы на физические устройства различных видов — серверы, рабочие станции терминалы, принтеры, маршрутизаторы и пр. — и связи между ними, представлен сетевыми и индивидуальными соединениями.	Диаграммы развертывания	ПК-10
71.	UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграммах состояний. Под деятельностью (англ. activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.	Диаграммы деятельности	ПК-9

	Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.		
72.	диаграмма, на которой изображаются взаимодействия между частями композитной структуры или ролями кооперации. В отличие от диаграммы последовательности, на диаграмме коммуникации явно указываются отношения между объектами, а время как отдельное измерение не используется (применяются порядковые номера вызовов).	Диаграмма коммуникации	ПК-10
73.	одна из разновидностей диаграммы деятельности в языке моделирования UML, которая может включать в себя также элементы диаграммы последовательности. Цель её создания ставится как увязывание в единое целое потока управления между узлами из диаграмм деятельности с последовательностью сообщений между линиями выполнения диаграмм последовательности. Расширение синтаксиса осуществляется за счёт использования ссылок на взаимодействия, которые основаны на диаграмме последовательности. Основным назначением диаграммы обзора взаимодействий является создание высокоуровневого представления об общем характере взаимодействий в проектируемой системе. Она не концентрирует своё внимание на отдельных аспектах работы так, как это происходит в других видах	Диаграмма обзора взаимодействия	ПК-7

	поведенческих диаграмм (коммуникации, последовательности, взаимодействия и т.п.) При её конструировании она компонуется с использованием нотации диаграммы последовательности, однако обозначения ветвлений и разделений заимствуются из диаграммы деятельности.		
74.	предусматривающий, что требования к созданию, функционированию и развитию ПО определяются со стороны более сложной, включающей его в себя системы	Принцип включения при создании и развитии программного обеспечения	ПК-9
75.	состоящий в том, что на всех стадиях создания, функционирования и развития ПО его целостность будет обеспечиваться связями между подсистемами, а также функционированием подсистемы управления	Принцип системного единства при создании и развитии программного обеспечения	ПК-9
76.	предусматривающий в ПО возможность его наращивания и совершенствования компонентов и связей между ними	Принцип развития при создании и развитии программного обеспечения	ПК-10
77.	закрывающийся в том, что ПО обеспечивает связанность обработки информации как отдельных элементов, так и всего объема данных в целом на всех стадиях обработки	Принцип комплексности при создании и развитии программного обеспечения	ПК-9
78.	т. е. во всех, подсистемах, средствах обеспечения и компонентах ПО используются единые термины, символы, условные обозначения и способы представления	Принцип информационного единства при создании и развитии программного обеспечения	ОПК-2
79.	состоящий в том, что язык, символы, коды и средства программного обеспечения согласованы, обеспечивают совместное функционирование всех подсистем и сохраняют открытой структуру системы в целом	Принцип совместимости при создании и развитии программного обеспечения	ПК-10

80.	предопределяющий, что подсистемы и компоненты ПО инвариантны к обрабатываемой информации, т. е. являются универсальными или типовыми	Принцип инвариантности при создании и развитии программного обеспечения	ОПК-2
81.	— процедурно-ориентированные — алгоритмы; — объектно-ориентированные — классы и объекты; — логически-ориентированные — цели, выраженные в исчислении предикатов; — ориентированные на правила — правила «если..., то...»; — ориентированные на ограничения — инвариантные соотношения; — параллельное программирование — потоки данных.	Основные парадигмы программирования	ПК-9
82.	Под спецификацией понимается достаточно полное и точное описание решаемой задачи на этапах проекта. Спецификация является моделью проектируемого объекта (программы). Спецификация может описывать соглашение между программистами и заказчиками (пользователями). Программист берется написать программу, а пользователь соглашается не полагаться на знания о том, как именно эта программа реализована, т. е. не предполагать ничего такого, что не было бы указано в спецификации. Такое соглашение позволяет разделить анализ реализации от собственно использования программы. Спецификации дают возможность создавать логические основы, позволяющие успешно «разделять и властвовать».	Понятие спецификации	ПК-7

83.	не является универсальным. Довольно сложно реализовать с использованием диалога данного типа ввод значений в широком диапазоне. Вертикальные меню предпочтительнее горизонтальных. Нежелательно в отдельных меню предлагать более семи тем. Меню с большим количеством тем желательно представить системой иерархической системы подменю. Часто используется в программах меню в стиле фирмы "Lotus" из одного главного горизонтального меню и иерархически построенных остальных вертикальных подменю. Строка единственного главного горизонтального меню придает уверенность начинающему пользователю путем сообщения ему факта о том, что имеется подменю в программе.	Диалог программы типа меню	ПК-11
84.	отображает на отдельном экране какой-то бланк. Посредством клика мыши или нажатия клавиш <Tab>, а также клавиш-стрелок перемещения курсора обеспечивается подвод курсора на любое из выделенных полей ввода информации. Обычно при заполнении какого-либо поля бланка осуществляется контроль кодов нажатых клавиш (ввод по маске). Трудность реализации диалога состоит в принятии решения по факту ввода недопустимой по значению информации в поле или ввода недопустимой совокупности значений в ряд полей.	Диалог программы типа заполнения бланков	ОПК-2
85.	Сложность использования диалога данного типа заключается в необходимости предварительного изучения пользователем языка команд и возможных	Диалог программы на основе команд	ПК-11

	последовательностей команд. Если команд мало в списке возможных команд, то пользователю по запросу "?" может быть выдан список возможных команд. После набора имени команды возможна выдача списка полей команды и их назначение.		
86.	позволяет на экране монитора организовать совокупность окон как отдельных программ, так и отдельных документов и осуществлять работу в каждом из отдельных окон. Данный диалог необходим специалистам, "у которых стол завален бумагами", т. е. специалистам, которым для получения нового документа необходима информация сразу из нескольких документов.	Диалог программы типа работы в окнах	ОПК-5
87.	отображает информацию в двухмерной системе координат по принципу игры "в морской бой" и в режиме "что вижу, то и получу в распечатке" позволяет придать табличной информации форму, необходимую пользователю	Диалог программы типа по принципу электронной таблицы	ПК-11
88.	обеспечивает пользователю просмотр на экране монитора текстовой (графической, видео, аудио и др.) информации с возможностью получения дополнительной информации при выборе пользователем выделенных на экране ключевых слов (ссылок).	Диалог программы типа гипертекста	ОПК-2
89.	обычно обеспечивает выдачу пользователем запросов на ограниченном естественном языке. Наибольшую сложность при реализации диалога данного типа представляет составление ограниченного тезауруса слов запроса и изучение данного тезауруса пользователем.	Диалог программы приближения к естественному языку	ОПК-2

90.	используется в различных тренажерах и может основываться на использовании особого оборудования типа кибершлем, тактильные перчатки, система запахов и т. д.	Диалог программы типа виртуальная реальность	ОПК-2
91.	Под структурой данных программ в общем случае понимают множество элементов данных, множество связей между ними, а также характер их организованности. Под организованностью данных понимается продуманное устройство с целью рационального использования по назначению. Примеры организованности данных: стек, организованный массивом; структура данных для хранения информации о студентах; файл, имеющий организацию текстового файла, байтная организация физической памяти машины. Н. Вирт определил понятие программы следующим образом: Алгоритмы + структуры данных = программы.	Структура данных программ	ОПК-2
92.	Понятие "физическая структура данных" отражает способ физического представления данных в памяти машины и называется еще структурой хранения, внутренней структурой, структурой памяти или дампом. Рассмотрение структуры данных без учета ее представления в машинной памяти называют абстрактной, или логической, структурой данных. В общем случае между логической и соответствующей ей физической структурами имеется различие, вследствие которого существуют правила	Физическая структура данных	ПК-7

	отображения логической структуры на физическую структуру. Структуры данных, применяемые в алгоритмах, могут быть чрезвычайно сложными. В результате выбор правильного представления данных часто служит ключом к удачному программированию и может в большей степени сказываться на производительности программы, чем детали используемого алгоритма.		
93.	то деятельность, направленная на обеспечение необходимых условий для работы коллектива разработчиков программного обеспечения (ПО), на планирование и контроль деятельности этого коллектива с целью обеспечения требуемого качества ПО, выполнения сроков и бюджета разработки ПО. Часто эту деятельность называют также управлением программным проектом (software project management). Здесь под программным проектом (software project) понимают всю совокупность работ, связанную с разработкой ПО, а ход выполнения этих работ называют развитием программного проекта	Управление разработкой программных систем	ПК-7
94.	— составление плана-проспекта по разработке ПО; — планирование и составление расписаний по разработке ПО; — управление издержками по разработке ПО; — текущий контроль и документирование деятельности коллектива по разработке ПО; — подбор и оценка персонала коллектива разработчиков ПО.	Общие процессы (виды деятельности) по управлению разработкой ПО	ПК-10

95.	включает формулирование предложений о том, как выполнять разработку ПО. Прежде всего должно быть зафиксировано, для кого разрабатывается ПО: • для внешнего заказчика; • для других подразделений той же организации; • является инициативной внутренней разработкой. В плане-проспекте должны быть установлены общие очертания работ по созданию ПО и оценена стоимость разработки, а также предоставляемые для разработки ПО материально-финансовые ресурсы и временные ограничения. Кроме того, он должен включать обоснование, какого рода коллективом должно разрабатываться ПО (специальной организацией, отдельной бригадой и т. п.). И наконец, должны быть сформулированы необходимые технологические требования (включая, возможно, и выбор подходящей технологии программирования).	Составление плана-проспекта по разработке ПО	ОПК-5
96.	это деятельность, связанная с распределением работ между исполнителями и по времени их выполнения в рамках намеченных сроков и имеющихся ресурсов.	Планирование и составление расписаний по разработке ПО	ПК-7
97.	это деятельность, направленная на обеспечение подходящей стоимости разработки в рамках выделенного бюджета. Она включает оценивание стоимости разработки проекта в целом или отдельных его частей, контроль выполнения бюджета, выбор подходящих вариантов его расходования. Эта деятельность тесно	Управление издержками по разработке ПО	ПК-11

	связана с планированием и составлением расписаний в течение всего периода выполнения проекта. Основными источниками издержек являются затраты на аппаратное оборудование (hardware), вербовку и обучение персонала, оплату труда разработчиков		
98.	это непрерывный процесс слежения за ходом развития проекта, сравнения действительных состояния и издержек с запланированными, а также документирования различных аспектов развития проекта. Этот процесс помогает вовремя обнаружить затруднения и предсказать возможные проблемы в развитии проекта.	Текущий контроль и документирование деятельности коллектива по разработке ПО	ОПК-2
99.	это деятельность, связанная с формированием коллектива разработчиков ПО. Имеющийся в распоряжении штат разработчиков далеко не всегда будет подходящим по квалификации и опыту работы для данного проекта. Поэтому приходится частично вербовать подходящий персонал и частично организовывать дополнительное обучение имеющихся разработчиков. В любом случае в формируемом коллективе хотя бы один его член должен иметь опыт разработки программных средств (систем), сопоставимых с ПО, которое требуется разработать. Это поможет избежать многих простых ошибок в развитии проекта.	Подбор и оценка персонала коллектива разработчиков ПО	ОПК-2
100.	Качество ПО формируется постепенно в процессе всей разработки ПО в каждой отдельной работе, выполняемой по программному проекту. Не могут вноситься	Обеспечение качества ПО	ПК-11

	изменения по улучшению качества в уже созданную программу. Для руководства этой деятельностью назначается специальный менеджер, подчиненный непосредственно директору, — менеджер по качеству. Ему непосредственно подчинены формируемые бригады по контролю качества. Для каждой работы организуется смотр (review) соответствующей бригадой. Смотру подлежат все программные компоненты и документы, включаемые в ПО, а также процессы их разработки. Смотр по контролю качества является функцией управления разработкой и связан с оценкой того, насколько результаты этой работы согласуются с декларируемыми требованиями относительно качества ПО.		
--	---	--	--