

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Дмитриев Николай Николаевич
Должность: Ректор
Дата подписания: 02.12.2024 10:48:49
Уникальный программный ключ:
f7c6227919e4cdbfb4d7b682991f8553b37cafb

Министерство сельского хозяйства Российской Федерации
Иркутский государственный аграрный университет
имени А.А. Ежевского

Колледж автомобильного транспорта и агротехнологий

УТВЕРЖДАЮ:
Директор



Н.Н. Бельков
«31» марта 2023 г

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРАКТИКЕ

ПП.06.01 ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

Специальность: 09.02.07 Информационные системы и программирование
(программа подготовки специалистов среднего звена)

Форма обучения: очная
3 курс, 6 семестр

Молодежный 2023

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Фонд оценочных средств для промежуточной аттестации по производственной практике, включает:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения (промежуточной аттестации) по производственной практике, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенции (ий).

2. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Рабочая программа производственной практики определяет перечень планируемых результатов обучения по производственной практике, соотнесенных с планируемыми результатами освоения образовательной программы.

Код	Наименование компетенции (планируемые результаты освоения ОП)	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенции
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	Умения: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и

		смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)
		Знания: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	Умения: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска Знания: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в	Умения: определять актуальность нормативно-правовой документации в профессиональной

	профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования Знания: содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде	Умения: организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности Знания: психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	Умения: грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе Знания: особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	Умения: описывать значимость своей специальности Знания: сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности
ОК 07	Содействовать сохранению	Умения: соблюдать нормы

	окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности Знания: правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	Умения: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение Знания: современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности
Профессиональные компетенции		
ПК 6.1.	Разрабатывать техническое задание на сопровождение информационной системы.	Практический опыт: Разрабатывать техническое задание на сопровождение информационной системы в соответствии с предметной областью. Умения: Поддерживать документацию в актуальном состоянии. Формировать предложения о расширении функциональности информационной системы. <i>Дополнительно для квалификации "Специалист по информационным системам"</i> Формировать предложения о прекращении эксплуатации информационной системы или ее реинжиниринге. Знания: Классификация

		информационных систем. Принципы работы экспертных систем. Достижения мировой и отечественной информатики в области интеллектуализации информационных систем. <i>Дополнительно для квалификации "Специалист по информационным системам"</i> Структура и этапы проектирования информационной системы. Методологии проектирования информационных систем.
ПК 6.2.	Выполнять исправление ошибок в программном коде информационной системы.	Практический опыт: Исправлять ошибки в программном коде информационной системы в процессе эксплуатации. Осуществлять установку, настройку и сопровождение информационной системы. Умения: Идентифицировать ошибки, возникающие в процессе эксплуатации системы. Исправлять ошибки в программном коде информационной системы в процессе эксплуатации. Знания: Основные задачи сопровождения информационной системы. Регламенты и нормы по обновлению и сопровождению обслуживаемой информационной системы.
ПК 6.3.	Разрабатывать обучающую документацию для пользователей информационной системы.	Практический опыт: Выполнять разработку обучающей документации информационной системы. Умения: Разрабатывать обучающие материалы для пользователей по эксплуатации ИС. Знания: Методы обеспечения и контроля качества ИС.

		Методы разработки обучающей документации.
ПК 6.4.	Оценивать качество и надежность функционирования информационной системы в соответствии с критериями технического задания.	Практический опыт: Выполнять оценку качества и надежности функционирования информационной системы на соответствие техническим требованиям. Умения: Применять документацию систем качества. Применять основные правила и документы системы сертификации РФ. <i>Дополнительно для квалификации «Специалист по информационным системам»:</i> Организовывать заключение договоров на выполняемые работы. Выполнять мониторинг и управление исполнением договоров на выполняемые работы. Организовывать заключение дополнительных соглашений к договорам. Контролировать поступления оплат по договорам за выполненные работы. Закрывать договора на выполняемые работы. Знания: Характеристики и атрибуты качества ИС. Методы обеспечения и контроля качества ИС в соответствии со стандартами. Политику безопасности в современных информационных системах. <i>Дополнительно для квалификации «Специалист по информационным системам»:</i> Основы бухгалтерского учета и отчетности организаций Основы налогового законодательства Российской Федерации

ПК 6.5.	Осуществлять техническое сопровождение, обновление и восстановление данных ИС в соответствии с техническим заданием.	Практический опыт: Выполнять регламенты по обновлению, техническому сопровождению, восстановлению данных информационной системы. Организовывать доступ пользователей к информационной системе. Умения: Осуществлять техническое сопровождение, сохранение и восстановление базы данных информационной системы. Составлять планы резервного копирования. Определять интервал резервного копирования. Применять основные технологии экспертных систем. Осуществлять настройку информационной системы для пользователя согласно технической документации. Знания: Регламенты по обновлению и техническому сопровождению обслуживаемой информационной системы. Терминология и методы резервного копирования, восстановление информации в информационной системе.
---------	--	--

В рабочей программе производственной практики **ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ** определены тематическим планом.

3. ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

При проведении промежуточной аттестации в университете используются традиционные формы аттестации:

Форма промежуточной аттестации	Шкала оценивания
ЗАЧЕТ с оценкой	"неудовлетворительно", "удовлетворительно"

	"хорошо" "отлично"
--	-----------------------

Формой промежуточной аттестации по практике является дифференцированный зачет, в ходе которого обязательно учитываются виды работ, выполненные обучающимся во время практики, их объем и качество выполнения.

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ (ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ) ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ И (ИЛИ) ДЛЯ ИТОГОВОГО КОНТРОЛЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

4.1. Перечень вопросов к ЗАЧЕТУ с оценкой для оценивания результатов обучения в виде знаний и умений.

1. Методы организации работы в команде разработчиков. Системы контроля версий

Ответ: Методы разработки - авторская, коллективная, общинная. Системы контроля версий: Система контроля версий записывает все изменения, внесенные в файлы проекта, в специальную базу данных. Это позволяет вам легко вернуться к предыдущим версиям файлов или просмотреть историю изменений. Совместная работа: VCS позволяет нескольким разработчикам работать над одним и тем же проектом одновременно, объединяя их изменения и решая конфликты в случае необходимости.

Цели, задачи, этапы и объекты ревьюирования. Планирование ревьюирования

2. Цели, корректность и направления анализа программных продуктов. Выбор критериев сравнения. Представление результатов сравнения

Ответ: Большое количество программных продуктов на современном рынке зачастую вызывает вопросы пользователя: какое ПО выбрать, какое будет качественным, какое максимально отвечает потребностям? Вопросы о ПО возникают не только у пользователей, но и у программистов: как разработать качественное и востребованное ПО, что для того нужно сделать? Для ответов на эти вопросы используют два вида мыслительных операций, используемых для исследования объектов, — анализ и синтез.

3. Примеры сравнительного анализа программных продуктов

Ответ: Проведем сравнительный анализ данных программных продуктов. В качестве критериев для сравнительного анализа программных продуктов, выберем следующие:

- 1) A1- расчет нормативного расхода ГСМ и контроль экономии/перерасхода ГСМ;
- 2) A2 - надбавки на ГСМ в холодное время года, работа кондиционера или отопительной печи, работа специального оборудования (лебедка, свет, подъемник, т.п.);
- 3) A3 - отчеты по водителям (манера вождения каждого водителя) ;
- 4) A4 - отчеты по автотранспортному средству с учетом индивидуальных особенностей автотранспортного средства;
- 5) A5- визуализация отчетов(графики, диаграммы);

4. Цели, задачи и методы исследования программного кода

Ответ: Изучение и анализ качества кода: оценка чистоты, структурированности и легкости понимания кода. Улучшение читаемости и поддерживаемости кода: Создание кода, который легко понимать и изменять другим разработчикам. Поиск и устранение ошибок и дефектов: Обнаружение и исправление ошибок, которые могут привести к некорректной работе программы. Проверка соответствия стандартам и требованиям: Гарантировать соответствие кода установленным стандартам и требованиям проекта. Оптимизация производительности: Улучшение скорости работы программы и оптимизация использования ресурсов.

5. Механизмы и контроль внесения изменений в код

Ответ: Делать мелкие исправления в проекте можно путём непосредственной правки рабочей копии и последующей фиксации изменений прямо в главной ветви (в стволе) на сервере. Однако при выполнении объёмных работ такой порядок становится неудобным: отсутствие фиксации промежуточных изменений на сервере не позволяет работать над чем-либо в групповом режиме, кроме того, повышается риск потери изменений при локальных авариях и теряется возможность анализа и возврата к предыдущим вариантам кода в пределах данной работы. Поэтому для таких изменений обычной практикой является создание ветвей (branch), то есть «отпочковывание» от ствола в какой-то версии нового варианта проекта или его части, разработка в котором ведётся параллельно с изменениями в основной версии. Ветвь создаётся специальной командой. Рабочая копия ветви может быть создана заново обычным образом (командой извлечения рабочей копии, с указанием адреса или идентификатора ветви), либо путём переключения имеющейся рабочей копии на заданную ветвь.

6. Утилиты для review: обзор

Ответ: Самые популярные статические инструменты для код-ревью.

1. Review Board
2. Crucible
3. GitHub
4. Phabricator
5. Collaborator
6. CodeScene
7. Visual Expert
8. Gerrit
9. Rhodocode
10. Veracode
11. Reviewable
12. Peer Review для Tас

7. Предпроцессинг кода. Интеграция в IDE

Ответ: Система препроцессинга исходного кода представляет собой программное обеспечение, с помощью которого можно выполнить транспилицию, то есть преобразовать программу, написанную на одном языке программирования, называемым исходным, в эквивалентную программу на другом языке — целевом. Программа, преобразующая исходный код программы, написанной на одном языке программирования в качестве исходных данных, и производящий эквивалентный ему целевой код на другом языке программирования называется транспайлером. Программы, использующие такую технологию, также называют препроцессором, которые возвращают программный код с дополнительными изменениями, направленными на улучшение читаемости или ускорения работы по его написанию. Транслитируемым или исходным языком программирования считается язык программирования, в котором содержатся дополнительные операторы и который требует дополнительной обработки для их изменения. Исходный код на данном языке обрабатывается системой препроцессинга.

Интегрированная среда разработки (IDE) – это программное приложение, которое помогает программистам эффективно разрабатывать программный код. Оно повышает производительность разработчиков, объединяя такие возможности, как редактирование, создание, тестирование и упаковка программного обеспечения в простом для использования приложении. Так же как писатели используют текстовые редакторы, а бухгалтеры – электронные таблицы, разработчики программного обеспечения применяют IDE, чтобы упростить свою работу.

8. Валидация кода на стороне сервера и разработчика

Ответ: Валидация — это проверка верстки на корректность. Код считается валидным и качественным, если соответствует стандартам организации W3C и не содержит ошибок. Ошибкой может быть, например, некорректно составленный тег или не прописанный путь. По мере ввода, браузер и/или сервер проверяют данные, чтобы определить, соответствуют ли они требуемому формату. Валидация, выполняемая в браузере, называется валидацией на стороне клиента, а выполняемая на сервере — валидацией на стороне сервера.

9. Совместимость и использование инструментов ревьюирования в различных системах контроля версий

Ответ: Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием. К ним относятся:

- программы для работы с текстами (текстовые редакторы);
- графические редакторы;
- электронные таблицы (табличный процессор);
- системы управления базами данных (СУБД);
- программы подготовки презентаций;
- программы для Web-дизайна;
- программы распознавания символов;
- интегрированные среды.

Существуют кросс-компиляторы — компиляторы, генерирующие исполняемый код для платформы, отличной от той, на которой запущен сам компилятор.

10. Особенности ревьюирования в Linux. Настройки доступа

Ответ: Linux разделяет права доступа на три категории: пользователь, группа и другие. Каждой категории могут быть назначены свои разрешения на чтение, запись и выполнение. И все это мы можем изменить, если у нас есть на это право и прямое разрешение, с помощью утилиты `chmod`. Исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма. Основные виды прав доступа: Чтение файлов, папок, каталогов. Минимальные привилегии, разрешающие только просмотр объектов. Запись — полномочия, которые позволяют редактировать и дополнять существующие файлы. Как выставить права на папку "777"

1. Выделяете папку -> Files -> Change Attributes...
2. После чего устанавливаете права 777 и нажимаете «ОК».
3. Некоторые хостеры не поддерживают права 777, учтите и проверьте это.

Право доступа — это право на выполнение определенной операции над объектом. Например, право доступа `FILE_READ_DATA` определяет право на чтение из файла. Когда вы открываете дескриптор объекта, вы указываете набор прав доступа, соответствующих операциям, которые могут быть выполнены над объектом.

11. Типовые инструменты и методы анализа программных проектов

Ответ: Методы анализа информационных систем структурный подход, поддерживаемый методологией системного анализа и проектирования Structure Analysis and Design Technique

(SADT); объектно-ориентированный подход, поддерживаемый методологией Rational Unified Process (RUP) и языком моделирования Unified Modeling Language (UML). Некоторые из основных типов этих инструментов разработки программного обеспечения включают компиляторы, редакторы кода, дизайнеры GUI, отладчики и различные другие инструменты разработки и управления проектами. «Анализ» — это широкий термин, лучше всего квалифицированный, как в анализе требований (исследование требований) или анализе объектов (исследование объектов домена). Проектирование делает акцент на концептуальном решении, которое удовлетворяет требованиям, а не на его реализации. Анализ проекта (design review) — документированная, всесторонняя и систематическая проверка проекта с целью оценки его возможности выполнять требования к качеству, выявлять проблемы и определять способы их решения.

12. Инструментарий различных сред разработки

Ответ: Инструментальное программное обеспечение — программное обеспечение, предназначенное для использования в ходе проектирования, разработки и сопровождения программ. Обычно этот термин применяется для акцентирования отличия данного класса ПО от прикладного и системного программного обеспечения. Виды инструментального ПО

- Интегрированные среды разработки
- SDK.
- Компиляторы и кросс-компиляторы
- Интерпретаторы
- Линковщики
- Ассемблеры
- Отладчики
- Профилировщики.

Инструментальное программное обеспечение - программное обеспечение, используемое в ходе разработки, корректировки или развития других программ: редакторы, компиляторы, отладчики, вспомогательные системные программы, графические пакеты и др.

13. Измерительные методы оценки программ: назначение, условия применения

Ответ: Наряду с аналитическими методами для исследования и оценки параметров программ активно используются измерительные методы. Привлекательной стороной этих методов является их высокая достоверность. Поэтому они применяются для проверки имитационных и аналитических методов оценки характеристик программ по принципу: практика – лучший критерий истины.

В целом измерительные методы имеют следующее назначение:

1. Измерение параметров потребления программами ресурсов вычислительной системы с целью устранения дефектов производительности.
2. Предварительное измерение параметров системы для имитационных или аналитических моделей программ перед их последующим использованием. Это связано со сложностью оценки параметров моделей, особенно при использовании новых системных средств с неизвестными динамическими параметрами (например устройств ввода-вывода, нового процессора с не вполне известными характеристиками и т.п.).
3. Проверка адекватности имитационных или аналитических моделей и методов расчета характеристик выполнения программ по результатам моделирования.

Необходимые условия применения измерительных методов:

1. Наличие готовой программы, подлежащей измерительному исследованию;
2. Наличие реальной вычислительной системы (а не её модели) для прогона программы;
3. Наличие аппаратных или программных средств проведения измерений;
4. Создание условий снижения искажений, вносимых в функционирование системы в процессе проведения измерений, до приемлемого уровня.

14. Корректность программ. Эталоны и методы проверки корректности

Ответ: Эталоны для проверки корректности программ могут использоваться в следующих трех формах: 1. Формализованные правила. 2. Программные спецификации. 3. Тесты.

Формализованные правила - имеют достаточно неопределенностей, так как определяются двумя видами требований:

- требования стандартов (общероссийских и стандартов предприятий);
- требования языков и технологий программирования.

В зависимости от стадии и характера проверки разделяются тесты: делятся на статические и динамические. Статическое тестирование - ручное тестирование программ, начиная со стадии формирования требований к программе. На стадии кодирования при статическом тестировании некоторую часть маршрутов исполнения тестируют вручную. Динамическое тестирование подразумевает достаточно полную структурную и функциональную проверку выполнения программы.

Как формируются эталоны для тестирования? Существует несколько способов формирования эталонов:

1. Использование аналитических выражений. Этот способ особенно подходит при детерминированном тестировании, так как имеется возможность сравнить результаты тестирования с ожидаемыми результатами. Имеются ограничения в использовании этого метода, если неизвестны или отсутствуют аналитические выражения связывающие входные данные и результаты; иногда требуется использовать много допущений.
2. Использование моделирования на ЭВМ. Способ является универсальным. При этом ряд данных моделируется другим способом и по другим алгоритмам, нежели испытываемая программа и на других ЭВМ. Причем наборы входных данных создаются по случайным законам, что обеспечивает высокую гибкость этого способа.
3. Использование результатов испытаний предшествующих вариантов программ.

При этом используется ранее накопленный опыт испытателя или других исследователей, выраженный в экспертных оценках ожидаемых результатов.

Степень достоверности проверки корректности программ при использовании этих методов убывает по номерам способов формирования эталонов.

В 1-ом случае обеспечивается 100% гарантия корректности программ, в третьем случае такой уверенности нет, но мы можем убедиться в том, что программа работает так же или иначе, чем аналогичный вариант. Менее достоверные тесты приходится использовать из-за недостаточности сил и средств.

15. Метрики, направления применения метрик. Метрики сложности. Метрики стилистики

Ответ: Виды метрик. Метрики продукта разделяют на внешние и внутренние. Первые отражают потенциал компании на фоне конкурентов и уровень лояльности покупателей, вторые — продуктивность деятельности. При оценке сложности программ, как правило, выделяют три основные группы метрик: метрики размера программ, метрики сложности потока управления программ и метрики сложности потока данных программ.

Метрика — показатель для оценки работы маркетинга, продаж и бизнеса в целом. С помощью метрик можно понять, например, приносит ли прибыль реклама, сколько новых клиентов получила компания за последний месяц, какой процент покупателей совершает повторные покупки. Метрика программного обеспечения (англ. software metric) — мера, позволяющая получить численное значение некоторого свойства программного обеспечения или его спецификаций. Метрики являются важным инструментом для оценки эффективности организации. Метрики ввода, метрики процесса, метрики вывода, метрики результата и метрики воздействия — все они играют свою роль. Иногда они действуют как опережающие индикаторы; иногда они используются как запаздывающие индикаторы. Общие метрики

включают: Цикломатическая сложность — количество путей, возможных в системе. Высокая цикломатическая сложность означает, что путей слишком много, что затрудняет понимание и поддержку. Метрики Холстеда — объем и сложность понимания кодовой базы системы.

Для чего рассчитываются метрики стилистики? позволяет определить основные умственные затраты на создание программы. Процесс создания программы условно можно представить как ряд операций: 1) осмысление предложения известного алгоритма; 2) запись предложения алгоритма в терминах используемого языка программирования.

16. Исследование программного кода на предмет ошибок и отклонения от алгоритма

Ответ: Статический анализ кода (англ. static code analysis) — анализ исходного кода программного обеспечения, производимый без реального выполнения исследуемых программ (в отличие от динамического анализа). Инструмент статического анализа проверяет такие языки программирования, как Java, C/C++, Python и другие, на предмет уязвимостей, ошибок и соответствия стандартам кодирования. Это обеспечивает улучшение качества программного обеспечения. Рецензирование кода, обзор кода, ревизия кода (англ. code review) или инспекция кода (англ. code inspection) — систематическая проверка исходного кода программы с целью обнаружения и исправления ошибок, которые остались незамеченными в начальной фазе разработки. Ошибки в программах бывают:

- логическими;
- синтаксическими;
- взаимодействия;
- компиляционные;
- ресурсные;
- арифметические;
- среды выполнения.

17. Программные измерительные мониторы

Ответ: Программные измерительные мониторы (ПИМ) — это совокупность команд или программ, выполняемых исключительно с целью проведения измерений. Обычно ПИМ — это специальные программные средства, под управлением которых производится выполнение программы на той же ЭВМ, на которой измеряемая программа и должна выполняться. При этом ПИМ собирает данные о ходе выполнения программ и накапливает их в памяти.

То, что эти команды ПИМ должна выполнять сама измеряемая система, приводит к возникновению искажений. Количество искажений зависит от частоты обнаруживаемых событий и от операций, выполняемых измерителем при обнаружении каждого события.

Идея использовать измерительные средства в процессе разработки программ — не нова. В настоящее время существует несколько типов таких программных средств. Особое место среди них занимают профилировщики. Профилировщики (называемые также анализаторами процесса выполнения программ) — это программы, позволяющие получить ряд количественных данных о процессе выполнения объекта разработки (снять профиль разрабатываемой программы) и на основании этих данных выявить в программе "узкие места", отрицательно сказывающиеся на эффективности ее работы. Профиль программы может содержать, например, следующую информацию о процессе выполнения программы:

- как и на что расходуется время работы программы;
- сколько раз выполняется данная строка программы;
- сколько раз и какими модулями вызывается данный модуль программы.

Один из самых известных профилировщиков — Turbo Profiler фирмы Borland International. Система Turbo Profiler контролирует каждый шаг выполнения программы и предоставляет подробные статистические сведения о всех этапах ее работы. Такой подход к измерениям

позволяет проводить профилирование в интерактивном режиме и получать большое количество сведений о работе программы, но вносит большие временные искажения в процесс ее выполнения. Кроме того, для проведения измерительного эксперимента тестируемая программа должна быть откомпилирована с отладочной информации при отключении оптимизации компилятора. Это делает профилирование неадекватным реальному выполнению программы. Для получения достоверной информации о времени выполнения от программиста требуется достаточно высокая квалификация в методике планирования и проведения измерительного эксперимента. Таким образом, Turbo Profiler ориентирован прежде всего на профессиональных программистов. За счет грамотного планирования эксперимента и отказа от ряда сервисных функций можно повысить точность временных измерений до 55 мс. Тем не менее, такая точность в ряде случаев недостаточна.

18. Защита программ от исследования

Ответ: Противодействие попыткам запуска и/или исполнения защищенной программы обязательно необходимо сопровождать механизмами, предотвращающими возможность исследования программы, в том числе и зарегистрированных (законных) копий. Повторим, что для взлома защиты нарушителю прежде всего необходимо найти защитный механизм и понять логику его работы, то есть исследовать программу.

Средства исследования программ

Основными инструментами для исследования программ являются дисассемблеры и отладчики.

Дисассемблирование - это получение из исполняемого кода программы код на языке ассемблера.

Дисассемблер - программа, осуществляющая дисассемблирование.

Интерактивный дисассемблер - программа, тесно взаимодействующая с пользователем в процессе дисассемблирования.

Отладчик - программа, предназначенная для анализа поведения другой программы, обеспечивающая остановку в указанных точках и позволяющая просматривать (редактировать) содержимое ячеек памяти, регистров процессора и команды программы.

Эмулирующий отладчик - отладчик, который самостоятельно интерпретирует и выполняет команды программы (без использования реального процессора).

Существует также множество программ-утилит, предназначенных для вспомогательных операций по изучению логики работы механизма защиты. Широко используются

- шестнадцатеричные просмотрщики - редакторы;
- редакторы таблиц экспорта/импорта;
- так называемые файловые мониторы, позволяющие отслеживать операции работы с файлами;
- а также мониторы реестра, создающие протокол обращений к реестру
- и многие другие.

Например, с помощью файлового монитора (FileMonitor) взломщик может отследить работу защищенной программы с файлами и обнаружить ключ (пароль), хранящийся в некотором файле. Произведя анализ протокола обращений к реестру с помощью монитора реестра (RegMon), взломщик может обнаружить ключ (пароль), хранящийся в системной базе данных Registry.

Инструментарий современного хакера настолько развит, что все попытки авторов защит противодействовать исследованию программ с точки зрения высококвалифицированных хакеров считаются безрезультатными.

С помощью современных версий интерактивных дисассемблеров и эмулирующих отладчиков может быть обнаружена практически любая защита.

! Тем не менее отказываться от использования приемов и методов защиты от дисассемблирования кода программы и ее работы под отладчиком нельзя.

Напомним, что абсолютной защиты вообще не бывает. Эффективной защитой считается такая, на взлом которой необходимы материальные и трудовые затраты, во много раз превышающие затраты на покупку программного обеспечения. Поэтому затруднение взлома защиты любыми путями и методами является оправданным. Если преодолеть защиту не сможет молодой неопытный взломщик, и заказчику придется обращаться к высококвалифицированному специалисту, - это уже плюс.

Считая защиту от дисассемблирования и отладки, рассчитанной на взломщика средней квалификации, назовем такую защиту затруднением анализа программ. Рассмотрим ее основные моменты.

20 Исследование кода вредоносных программ

Ответ: Вредоносное ПО — это приложения или код, которые препятствуют нормальному использованию конечных устройств. Когда устройство заражено вредоносным ПО, вы можете столкнуться с несанкционированным доступом, компрометацией данных или блокировкой и требованием заплатить выкуп. Вы также можете проверить их наличие вручную:

1. Откройте на устройстве приложение "Настройки".
2. Выберите Безопасность и конфиденциальность Система и обновления. Если вас интересуют исправления уязвимостей, нажмите Обновление системы безопасности. ...
3. Следуйте инструкциям на экране.

Разновидности вредоносного кода

- Вирусы Вирус – это самовоспроизводящийся вредоносный код, который прикрепляется к программам с поддержкой макросов для активации. ...
- Черви ...
- Троянские программы ...
- Межсайтовый скриптинг (XSS) ...
- Бэкдоры ...
- Троянец Emotet. ...
- Червь Stuxnet

Обнаружение вредоносного ПО. Домашняя страница вашего браузера меняется или автоматически перенаправляет вас на сайты, которые вы не собирались посещать. Вы замечаете новые панели инструментов браузера или дополнительные программы. Вы видите много всплывающих объявлений, включая неуместную рекламу или рекламу на сайтах, которые вы не ожидаете увидеть, например, на правительственных веб-сайтах.

Приложение 1

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Иркутский государственный аграрный университет имени А.А. Ежевского
Колледж автомобильного транспорта и агротехнологий

Согласовано:
Руководитель практики
от предприятия _____
/ _____ /
(ФИО руководителя)
« ____ » _____ 20__ г.

**ОТЧЕТ
ПО ПРАКТИКЕ
ПП.03.01 ПРОИЗВОДСТВЕННАЯ ПРАКТИКА**

ПМ.03 Ревьюирование программных модулей

Студента _____ (ФИО)

Курса _____ группы _____

с «__» _____ 20__ г. по «__» _____ 20__ г.

Руководитель практики от колледжа

Руководитель практики от предприятия

Молодежный 20__

Приложение 2

Образец аттестационного листа по практике (формат А 4)

ФГБОУ ВО ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ

ИМЕНИ А.А ЕЖЕВСКОГО

КОЛЛЕДЖ АВТОМОБИЛЬНОГО ТРАНСПОРТА И АГРОТЕХНОЛОГИЙ

АТТЕСТАЦИОННЫЙ ЛИСТ ПО ПРАКТИКЕ

обучающ____ся группы _____ специальность СПО _____

____, прошел (ла) учебную/производственную практику по ПМ _____

с _____ 201__ г. по _____ 201__ г. в организации _____

Оценка уровня освоения профессиональных компетенций

Наименование профессиональных компетенций		Уровень освоения*	Примечание
ПК 3.1.			
ПК 3.2.			
ПК 3.3.			
ПК 3.4.			

*Оценивание осуществляется по пятибалльной системе путем выделения оценки.

В случае неявки обучающегося на практику используется оценка 0.

Дата «____» _____ 201__ г.

Подпись руководителей практики:

от колледжа _____ / _____ /
расшифровка подписи

от организации _____

М.П. _____
расшифровка подписи

Приложение 3

ПРОИЗВОДСТВЕННАЯ ХАРАКТЕРИСТИКА

на _____, обучающего(ую)ся

_____ группы _____

специальность СПО _____

в период производственной практики по ПМ _____

с _____ 201__ г. по _____ 201__ г. в организации _____

За период практики обучающийся выполнял следующие виды работ:

_____, которые

соответствуют _____
(квалификации)

отношение обучающегося – практиканта к выполняемой работе _____

_____ степень выполнения поручений

качественный уровень и степень подготовленности обучающегося к самостоятельному выполнению отдельных заданий _____

дисциплинированность и деловые качества _____

наличие отрицательных черт, действий, проявлений, характеризующих практиканта с негативной стороны _____

Оценка уровня освоения общих компетенций

Наименование общих компетенций		Уровень освоения*	Примечание
ОК 01.			
ОК 02.			
ОК 09.			

*Оценивание осуществляется по пятибалльной системе путем выделения оценки. В случае неявки обучающегося на практику используется оценка 0.

Рекомендуемая оценка по практике _____

Дата «___» _____ 201__ г.

Подпись руководителей практики: от организации _____
_____ расшифровка подписи

Подпись руководителя колледжа от организации _____
М.П. _____ расшифровка подписи

Приложение 4

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Иркутский государственный аграрный университет им. А.А.Ежевского
Колледж автомобильного транспорта и агротехнологий

ДНЕВНИК

Производственной практики по профилю специальности

Молодежный _____

Инструктаж по технике безопасности

Вводный инструктаж

Провел _____ Подпись _____ Дата _____	Инструктаж получил и освоил _____ Подпись _____ Дата _____
---	--

Первичный инструктаж на рабочем месте

Провел _____ Подпись _____ Дата _____	Инструктаж получил и освоил _____ Подпись _____ Дата _____
---	--

Разрешение на допуск к работе

Разрешено допустить к самостоятельной работе

Дата _____ 20 ____ г.

Начальник цеха (отдела) _____

Подпись _____

Замечание лиц, осуществляющих контроль за прохождением практики

Дата	Замечание	Подпись

Студент _____ / _____ /

Разработчик: доцент Федурин Н.И.



ФОС одобрен на заседании предметно-цикловой комиссии технических дисциплин, протокол № 8 от «29» марта 2023 г.

Председатель ПЦК


(подпись)

Е.А. Хуснудинова

(И.О. Фамилия)

СОГЛАСОВАНО:

Внешний эксперт:

Директор ИЭУПИ Иркутского Г.

доцент, к.т.н М.Н. Барсукова

