



Сетевой проект “О-ах, рекурсия!”

Исследование понимания смысла осторожности в программировании

Оценивание учебных потребностей студентов (учащихся)

Команда _____ Лягушатник _____

Вопрос: Где мы встречаемся с необходимостью быть осторожными в программировании?

Пройдите Тест

Наша таблица З - Х - Н

ФИО или имя и первая буква фамилии	Что мы Знаем об осторожности в программировании	Что мы Хотим узнать об осторожности в программировании	Что мы Узнали (чему и как Научились) об осторожности в программировании (заполняется в конце проекта)
Полуян Максим Владимирович	Валидация данных: Проверка вводимых данных для предотвращения уязвимостей. Обработка исключений: Правильное управление ошибками. Тестирование: Проведение юнит-тестов и других проверок.	Методы валидации данных: Как эффективно проверять и обрабатывать вводимые данные. Стратегии обработки исключений: Какие подходы использовать для управления ошибками. Практики тестирования: Как	Нужно быть аккуратным при создании программы и пытаться найти её слабые места, чтобы улучшить код.

	Код-ревью: Проверка и улучшение кода другими разработчиками. Версионный контроль: Использование систем, таких как Git. Безопасность кода: Применение практик для защиты данных. Документирование: Подробное описание кода и архитектуры проекта.	внедрить юнит-тесты и другие методы проверки. Код-ревью: Как проводить проверки и улучшения кода. Безопасность программного обеспечения: Как защитить код и данные от угроз.	
Шевкопляс Владимир Сергеевич	Проверяйте входные данные; Обрабатывайте исключения; Документируйте код; Тестируйте программу; Следуйте стандартам кодирования.	Как предосторожность влияет на качество кода? Какие методы и практики помогают избежать ошибок в программировании? Как лучше всего тестировать и документировать код? Какие стандарты и инструменты следует использовать для безопасного программирования?	Необходимо делать код легко читабельным для вычисления ошибок и возможности доработки кода
Шестаков Александр Александрович	Осторожность в программировании включает в себя ряд практик и принципов, направленных на предотвращение ошибок, уязвимостей и недоразумений в коде. Первое и одно из наиболее	Для этого мы хотим узнать какие существуют наиболее эффективные и современные методы для обеспечения осторожности в программировании. °Инструменты и технологии: Какие инструменты могут помочь в автоматизации проверки кода на наличие ошибок и уязвимостей?	Требуется хорошо понимать то, с чем работаешь в программировании для избежания проблем и непонимания

	важного это проверка входных данных: Важно всегда проверять данные, поступающие от пользователей или других систем, чтобы предотвратить атаки. Второе это обработка ошибок: Программисты должны правильно обрабатывать ошибки и исключения, чтобы избежать краха приложения и предоставить пользователям понятные сообщения об ошибках. Третье это чистый и читаемый код: Следование принципам чистого кода и хорошей архитектуры помогает другим разработчикам легче понимать и поддерживать код, что снижает вероятность ошибок. Четвёртое это то, что следует тестировать программу, так как это помогает выявлять ошибки на ранних стадиях разработки. Пятое безопасность: Учитывать аспекты безопасности на всех этапах разработки, включая шифрование данных и использование безопасных протоколов.	° Кейс-стадии: Как реальные компании внедряют осторожность в своих процессах разработки? Какие примеры успешного предотвращения инцидентов можно привести? ° Как писать безопасный код? ° Как правильно обрабатывать исключения и ошибки? ° Как писать код, который легко понимать другим разработчикам. Это включает в себя использование понятных имен переменных, комментариев и соблюдение стилей кодирования? ° Последнее и очень важное для программистов, это то, как писать эффективный код, который не только работает правильно, но и оптимально использует ресурсы? И ответить на все эти вопросы нам поможет проект, в котором мы участвуем.	
--	--	--	--

© Н.П. Макарова, 2020-2025