Титульник

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Учебная практика ПМ 02 «Ревьюирование программных модулей» занимает важное место в подготовке специалистов в области информационных технологий и программирования, обеспечивая формирование профессиональных компетенций, необходимых для разработки, оптимизации и оценки качества программного обеспечения. В ходе практики обучающиеся применяют теоретические знания на практике, развивают навыки самостоятельной работы и осваивают современные подходы к ревьюированию и тестированию программного кода. Основными целями практики являются систематизация, закрепление и расширение теоретических и практических знаний по выбранной специальности, развитие навыков самостоятельной работы и овладение методикой исследования при решении практических задач, а также определение уровня подготовки обучающегося и его способности применять знания для выполнения конкретных профессиональных задач.

Для достижения этих целей в рамках практики решаются задачи, связанные с изучением и работой с проектной документацией, разработанной с использованием графических языков спецификаций, выполнением оптимизации программного кода с применением специализированных инструментов, использованием методов и технологий тестирования и ревьюирования кода и документации, применением стандартных метрик для прогнозирования затрат, сроков и качества разработки, изучением принципов планирования и контроля развития программного проекта, а также современных стандартов качества программных продуктов. Кроме того, важной задачей является оформление отчета по учебной практике в соответствии с установленными нормативными требованиями.

Ревьюирование программного кода представляет собой процесс систематической проверки исходного кода, направленный на выявление ошибок, повышение качества программного продукта и обеспечение его соответствия стандартам разработки. Этот процесс включает анализ кода на предмет корректности, читаемости, производительности и безопасности, который может выполняться как индивидуально, так и в команде, с использованием автоматизированных инструментов или вручную. Ревьюирование играет ключевую роль в разработке программного обеспечения, поскольку позволяет выявлять и устранять ошибки на ранних стадиях, снижая затраты на исправление дефектов в дальнейшем. Оно способствует улучшению читаемости и поддерживаемости кода, упрощая его сопровождение, а также обеспечивает обмен знаниями между участниками команды, что способствует их профессиональному росту. Кроме того, ревьюирование гарантирует соблюдение корпоративных и отраслевых стандартов качества и помогает снизить риски, связанные с уязвимостями и потенциальными проблемами, повышая надежность и безопасность программного обеспечения.

Таким образом, учебная практика ПМ 02 «Ревьюирование программных модулей» не только закрепляет теоретические знания, но и формирует практические навыки, необходимые для профессиональной деятельности в области разработки программного обеспечения. В рамках данной практики была выполнена задача по разработке класса «Дробное число со знаком» (Fractions) с реализацией арифметических операций и операций сравнения, а также проведено тестирование разработанных методов. Отчет по практике содержит описание выполненной работы, анализ результатов и выводы, подтверждающие достижение поставленных целей и задач.

Тема 1. Моделирование программного обеспечения

Моделирование программного обеспечения является ключевым этапом разработки, обеспечивающим структурированный подход к созданию программных решений. Выполнение задания по разработке класса «Дробное число со знаком» (Fractions) началось с анализа предметной области, который позволил определить основные требования и особенности задачи. Анализ предметной области включал знакомство с интегрированной средой разработки (IDE), изучение ее функций и возможностей, а также углубление знаний об основах объектно-ориентированного программирования (ООП) на языке C#.

На этапе знакомства с IDE была использована среда Microsoft Visual Studio, которая предоставляет широкий набор инструментов для написания, отладки и тестирования программного кода. Изучение функций IDE включало освоение возможностей автодополнения кода, встроенного отладчика, системы управления проектами и инструментов для анализа синтаксиса. Это позволило повысить эффективность работы над заданием и обеспечить удобство при написании и тестировании кода. Особое внимание уделялось настройке рабочего пространства и интеграции средств ревьюирования кода, что способствовало более качественной разработке.

Знание основ ООП на языке C# стало основой для проектирования класса «Дробное число со знаком». Были изучены ключевые принципы ООП, такие как инкапсуляция, обеспечивающая защиту данных класса, наследование, позволяющее расширять функциональность, и полиморфизм, обеспечивающий гибкость в реализации методов. Эти принципы были применены при создании класса, который должен представлять дробное число с двумя полями: целой частью в виде длинного целого числа со знаком и дробной частью в виде беззнакового короткого целого числа. Кроме того, задание требовало реализации арифметических операций (сложения, вычитания, умножения) и операций сравнения, что предполагало использование перегрузки операторов в C# для удобства работы с объектами класса.

Проектирование алгоритма решения задачи началось с формализации требований. Для представления дробного числа было решено использовать структуру класса, где целая часть хранится как переменная типа long, а дробная часть — как ushort. Алгоритмы арифметических операций были разработаны с учетом необходимости обработки знака числа и корректного вычисления результатов. Например, операция сложения предполагала учет знака целой части и суммирование дробных частей с возможным переносом в целую часть при превышении максимального значения дробной части. Аналогично, операции вычитания и умножения были спроектированы с учетом особенностей представления чисел. Для операций сравнения был разработан алгоритм, основанный на последовательном сравнении целых и дробных частей чисел. Все алгоритмы были задокументированы в виде псевдокода, что упростило последующую разработку программного кода.

Разработка кода на языке C# проводилась в соответствии с составленными алгоритмами. Был создан класс Fractions, включающий поля для целой и дробной частей, а также методы для выполнения арифметических операций и сравнения. Для реализации операций использовалась перегрузка операторов (+, -, \*, ==, !=, <, >, <=, >=), что позволило сделать код интуитивно понятным и удобным для использования. В функции main была реализована программа для тестирования всех методов класса. Тестирование включало создание объектов класса, выполнение арифметических операций и сравнений, а также вывод результатов для проверки корректности работы. Код был оптимизирован с учетом читаемости и производительности, а также прошел этап ревьюирования для выявления потенциальных ошибок и улучшения качества.

Таким образом, выполнение задания по моделированию программного обеспечения позволило не только разработать функциональный класс «Дробное число со знаком», но и углубить навыки работы с IDE, применения принципов ООП и проектирования алгоритмов. Проведенная работа подтвердила важность анализа предметной области и систематического подхода к разработке программного обеспечения, что является основой для создания надежных и качественных программных продуктов.