

Государственное бюджетное учреждение дополнительного
профессионального образования Республики Мордовия «Центр
непрерывного повышения профессионального мастерства педагогических
работников – «Педагог 13.ру»

Связь математики с программированием

Оркина Марина Александровна,
преподаватель ГБПОУ РМ «Зубово-
Полянский педагогический колледж»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Связь математики с программированием	4
2. Профессиональная направленность содержания обучения	6
3. Использование современных технологий и методик преподавания	8
4. Конкретный практический пример	10
5. Формирование профессиональных компетенций будущих специалистов	12
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

Программирование является одной из ключевых дисциплин современного образования, особенно в условиях развития цифровых технологий и информатизации всех сфер жизни. Однако оно тесно связано с различными областями знания, среди которых особое место занимает математика. Междисциплинарные связи между этими предметами позволяют углублять понимание обоих направлений, развивать критическое мышление и повышать уровень подготовки студентов.

Под программированием понимается процесс разработки программного обеспечения, включающий проектирование алгоритмов, написание исходного кода, тестирование и отладку программы. Для успешного освоения программирования необходимы базовые знания алгебры, геометрии, теории вероятностей и статистики, дискретной математики и математического анализа.

Математика представляет собой фундаментальную область науки, изучающую количественные отношения и пространственные формы реального мира. Она включает в себя арифметику, алгебру, геометрию, тригонометрию, теорию чисел, комбинаторику, топологию и многие другие направления.

Знание основ математики существенно повышает эффективность обучения программированию. Преподавателям рекомендуется интегрировать элементы математики в уроки программирования, предлагая студентам практические задания, направленные на развитие математических навыков.

1. Связь математики с программированием

Часто на вопрос «Зачем мне эта математика?» можно услышать ответ: «Чтобы развивать логику». Программирование – это та самая практическая область, где математика оживает и становится мощным инструментом для создания цифровых продуктов. Их связь не абстрактна, а конкретна и проявляется на каждом шагу.

Рассмотрим, как темы из общеобразовательного курса математики напрямую находят применение в программировании:

Алгебра и основы алгоритмизации

- Переменные и функции: понятие переменной $x=5$ в математике напрямую переносится в код. Математическая функция $f(x)=x^2+1$ становится функцией в программировании: `def f(x): return x**2 + 1.`

- Уравнения и неравенства: логика их решения (перенос слагаемых, смена знака) используется для составления условных операторов (`if a > b: ...`). Понимание систем уравнений помогает в решении задач, где несколько переменных взаимосвязаны.

- Координатная плоскость: любой элемент на экране (окно, кнопка, персонаж игры) имеет свои координаты (x, y) . Без понимания системы координат невозможно создать даже простейший интерфейс.

- Векторы и геометрические преобразования: движение объекта, его поворот, масштабирование – все это описывается с помощью векторов. Расчет траектории полета мяча в игре – это чистая геометрия.

- Производные и интегралы: для студентов эти понятия кажутся оторванными от жизни. Но в программировании они используются для создания плавных анимаций (расчет скорости изменения), в алгоритмах машинного обучения для поиска оптимальных решений (градиентный спуск) и в физических симуляторах.

- Случайность: генерация случайных чисел (`random.randint()` в Python) используется в играх (критический удар, выпадение лута), в симуляциях и для криптографии.

- Анализ данных: построение графиков и диаграмм, расчет среднего значения, дисперсии – все это программы делают на основе статистических формул, изученных в курсе математики.

Математика и программирование в общеобразовательном контексте не конкурируют, а усиливают друг друга:

- Математика дает «что» решать – она ставит задачу, предоставляет формальный язык и аппарат для ее описания.

- Программирование дает «как» решить – оно предоставляет инструмент для проверки гипотез, автоматизации расчетов и визуализации сложных понятий.

Например, решая систему уравнений вручную, студент понимает ее суть. Написав программу для решения любой системы уравнений, он закрепляет это понимание и видит его практическую мощь.

Общеобразовательный курс математики – это не набор скучных формул, а фундаментальный язык, на котором говорит технологический мир. Программирование же – это средство, позволяющее заставить этот язык работать. Понимая их глубинную связь, студент перестает просто «решать примеры», а начинает «создавать решения», что является главной целью современного образования.

Таким образом, изучение математики играет ключевую роль в формировании компетенций будущих программистов. Без понимания основных математических концепций невозможно эффективно разрабатывать программное обеспечение высокого качества. Именно поэтому включение элементов математики в курсы программирования способствует повышению уровня профессиональной подготовки студентов.

2. Профессиональная направленность содержания обучения

Обучение математике на первом курсе среднего профессионального образования (СПО) носит ярко выраженную профессионально-прикладную направленность. В отличие от школьной программы, где математика зачастую преподается как абстрактная наука, в СПО она становится инструментом для решения конкретных производственных и технических задач. Наиболее тесная и очевидная интеграция математического знания наблюдается с дисциплиной «Программирование». Эта взаимосвязь является не просто дополнением, а фундаментальной основой для формирования компетентного специалиста в целой группе технических и IT-направлений, например: 09.02.07 «Информационные системы и программирование», 09.02.02 «Компьютерные сети», 09.02.03 «Программирование в компьютерных системах», 15.02.08 «Технология машиностроения».

На первом курсе закладывается математический фундамент. Каждая тема имеет четкую профессиональную привязку. По требованиям нового ФГОС в курс математики включен модуль с практико-ориентированным содержанием. На примере специальности 09.02.07 «Информационные системы и программирование» в данном модуле изучают следующие темы, связанные с программированием:

1) Тема: «Основы математической логики».

Профессиональная направленность: булева алгебра – основа всех условных операторов (if/else), циклов и логических проверок в любом языке программирования.

Связь с программированием: составление сложных логических условий для управления потоком выполнения программы; построение таблиц истинности для отладки сложных условий.

2) Тема: «Теория множеств».

Профессиональная направленность: является теоретической основой для структур данных и реляционных баз данных. Операции над множествами

(объединение, пересечение, разность) напрямую реализованы в языках программирования (Python: set).

Связь с программированием: использование типов данных «множество» для эффективного решения задач (например, поиск общих друзей в социальной сети – это пересечение двух множеств).

3) Тема: «Элементы комбинаторики».

Профессиональная направленность: необходима для анализа производительности алгоритмов (сколько раз выполнится цикл?).

Связь с программированием: написание программ для генерации всех возможных перестановок, сочетаний и размещений для задач тестирования и решения головоломок.

4) Тема: «Основы теории графов».

Профессиональная направленность: графы – это модель для представления сетей (компьютерных, социальных), маршрутов (GPS-навигация), иерархий (структура сайта). Алгоритмы на графах (поиск в ширину, глубину, алгоритм Дейкстры) – это daily routine для программиста.

Связь с программированием: реализация структуры данных «граф» (списки смежности, матрица смежности) и написание алгоритмов для обхода графа или поиска кратчайшего пути.

Таким образом, содержание обучения математике на первом курсе СПО для IT и технических специальностей подчинено четкой цели: сформировать у будущего специалиста аппарат математического мышления, который он сможет непосредственно транслировать в код. Математика перестает быть абстракцией и становится рабочим инструментом. Интеграция с программированием позволяет не только глубже понять суть математических концепций через их практическую реализацию, но и сразу формировать профессиональные компетенции, востребованные на современном рынке труда.

3. Использование современных технологий и методик преподавания

Эффективная интеграция математики и программирования в системе среднего профессионального образования (СПО) не происходит сама по себе. Она требует целенаправленного использования современных педагогических технологий и активных методов обучения, которые превращают абстрактные математические понятия в практически значимые навыки. Целью такого подхода является формирование у студента комплексного «аналитико-алгоритмического» мышления, при котором он видит в математической задаче потенциальный алгоритм, а в алгоритме – ее математическую сущность.

Активные и интерактивные методы обучения смещают фокус с пассивного восприятия информации на активную деятельность студентов:

1) Проектная деятельность – это ключевой метод интеграции. Студентам предлагается не решать отдельные задачи, а выполнять сквозные проекты, например, проект «Калькулятор для статистического анализа». В рамках него студенты на математике изучают основы статистики: среднее арифметическое, дисперсию, среднеквадратическое отклонение; на программировании пишут функции на Python, объединяют эти функции в единое приложение с графическим интерфейсом.

2) Кейс-технологии (Case-based Learning). Студентам предлагаются проблемные ситуации (кейсы) из профессиональной сферы. Например, кейс «Оптимизация маршрута доставки». Дана карта (граф) с точками доставки. Задача: найти кратчайший путь. Математическая основа: теория графов. Программная реализация: создание структуры данных «граф» и написание алгоритма для поиска кратчайшего пути.

Геймификация значительно повышает мотивацию. Платформы Codewars, LeetCode (простые задачи), HackerRank предлагают задачи по алгоритмам, для решения которых необходимы знания математики

(комбинаторика, теория чисел). Студенты могут соревноваться в рейтинге. Проведение однодневного хакатона на тему «Математические алгоритмы в жизни» (например, создание алгоритма для шифрования сообщения или генерации фракталов) позволяет в сжатые сроки применить знания и проявить креативность.

Интеграция требует пересмотра подходов к преподаванию и оценке знаний. Преподаватель трансформируется из транслятора знаний в фасилитатора, наставника (ментора) и организатора проектной деятельности. Он помогает студентам находить связи между дисциплинами, направляет их в работе над проектами. Происходит смещение от оценки исключительно итоговых контрольных работ к формирующему оцениванию. Оценивается процесс: активность на занятиях, участие в обсуждении кейсов, качество кода в проекте, умение работать в команде.

Реализация интеграции математики и программирования на первом курсе СПО – это сложный, но необходимый процесс, требующий системного подхода. Он невозможен без внедрения современных образовательных технологий, которые ставят студента в центр учебного процесса, превращая его из пассивного слушателя в активного создателя. Использование проектных методов, интерактивных сред, геймификации создает ту самую профессионально-ориентированную среду, в которой у студента формируется не разрозненное знание, а целостная профессиональная компетенция. Такой подход не только повышает успеваемость, но и фундаментально готовит будущего специалиста к требованиям цифровой экономики.

4. Конкретный практический пример

Выпускник специальности «Информационные системы и программирование» часто занимается разработкой бизнес-приложений, систем аналитики и обработки данных. Предлагаемый проект имитирует реальную задачу, с которой он может столкнуться на рабочем месте.

Математическая составляющая: среднее арифметическое, медиана, дисперсия, среднеквадратическое отклонение, нормальное распределение, расчет вероятностей, построение гистограмм, графики распределения.

Программистская составляющая: язык Python, библиотеки NumPy, Matplotlib, Pandas, среда разработки Jupyter Notebook.

Практическое задание: «Разработать аналитический модуль для интернет-магазина».

Этап 1. Подготовка данных (базовое программирование). Связь с математикой: использование нормального распределения для генерации реалистичных данных.

Этап 2. Реализация статистических функций (интеграция математики и кода). Связь с математикой: прямая реализация математических формул в коде.

Этап 3. Применение функций и анализ результатов.

Этап 4. Визуализация результатов (прикладное применение).

Этап 5. Бизнес-аналитика (профессиональное применение).

Образовательные результаты:

1. Математические компетенции:

- понимание и применение формул статистики;
- интерпретация стандартного отклонения и дисперсии;
- работа с нормальным распределением;
- построение и анализ графиков.

2. Программистские компетенции:

- работа с массивами данных (NumPy);

- создание пользовательских функций;
- визуализация данных (Matplotlib);
- обработка и анализ данных (Pandas);
- структурирование кода.

3. Профессиональные компетенции:

- разработка аналитических модулей;
- создание бизнес-отчетов;
- визуализация данных для принятия решений;
- интерпретация статистических показателей в бизнес-контексте.

4. Методическое обеспечение:

- Jupyter Notebook как единая среда для кода, математических формул (через LaTeX) и выводов;
- пошаговое выполнение с немедленной визуальной обратной связью;
- возможность модификации параметров и наблюдения за изменением результатов.

Данный пример показывает, как абстрактные математические понятия становятся практическим инструментом в руках программиста, создающего полезные бизнес-приложения.

5. Формирование профессиональных компетенций будущих специалистов

1. Базовые теоретические компетенции.

- математическая грамотность – способность формализовать практические задачи в математические модели;
- алгоритмическое мышление – умение декомпозировать сложные проблемы на последовательные шаги;
- аналитический подход: формирование навыков анализа требований и проектирования решений.

Пример формирования: при изучении темы «Производная» студенты не просто заучивают формулы, а анализируют, как производная описывает скорость изменения процессов в реальных системах (скорость роста пользователей в приложении, изменение нагрузки на сервер).

2. Практические технические компетенции

- навыки программирования: умение реализовывать математические алгоритмы на языках высокого уровня;
- работа с данными: компетенции по обработке, анализу и визуализации данных;
- отладка и тестирование: навыки верификации математической корректности реализованных алгоритмов.

Пример формирования: создание программы для расчета оптимального маршрута доставки с использованием алгоритмов теории графов – от математической модели до рабочего прототипа.

3. Системные и проектные компетенции.

- системный подход: умение видеть задачу в контексте общей бизнес-логики;
- проектирование решений: способность выбирать оптимальные математические методы для конкретных задач;

– интеграционные навыки: компетенции по объединению математических модулей в комплексные системы.

Пример формирования: разработка модуля прогнозирования продаж, интегрируемого в существующую CRM-систему предприятия.

4. Методологические компетенции.

– исследовательские навыки: умение проводить сравнительный анализ алгоритмов;

– оптимизационное мышление: способность оценивать эффективность решений по времени и ресурсам;

– верификация результатов: навыки проверки достоверности и точности вычислений.

Пример формирования: Сравнение эффективности различных алгоритмов сортировки с математическим обоснованием их временной сложности.

5. Коммуникативные и управленческие компетенции.

– техническая коммуникация: умение объяснять математические концепции нетехническим специалистам;

– документирование: навыки создания технической документации с математическим обоснованием решений;

– проектное управление: базовые компетенции по планированию и оценке сроков реализации.

Пример формирования: Подготовка презентации для заказчика с обоснованием выбранного алгоритма шифрования данных на основе математической стойкости.

6. Адаптационные и инновационные компетенции.

– самообучение: способность самостоятельно осваивать новые математические методы и технологии;

– критическое мышление: умение оценивать применимость различных подходов к решению задач.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Далингер, В. А. Избранные вопросы информатизации школьного математического образования: монография / В. А. Далингер ; науч. ред. М. П. Лапчик. - 4-е изд., стер. - Москва: Флинта, 2021. - 150 с. - ISBN 978-5-9765-1159-0. - Текст: электронный. - URL: <https://znanium.ru/catalog/product/1843166> (дата обращения: 19.09.2025)

2. Математика и информатика: учебник и практикум для среднего профессионального образования / под редакцией В. Д. Элькина. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2025. — 402 с. — (Профессиональное образование). — ISBN 978-5-534-10683-1. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583918> (дата обращения: 19.09.2025).

3. Факторович, А. А. Педагогические технологии: учебник для среднего профессионального образования / А. А. Факторович. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2025. — 128 с. — (Профессиональное образование). — ISBN 978-5-534-13194-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588107> (дата обращения: 19.09.2025).

4. Федеральные государственные образовательные стандарты (ФГОС) СПО по специальностям: 09.02.07, 09.02.02, 09.02.03, 15.02.08. – Режим доступа: <https://fgos.ru/> (дата обращения: 05.09.2025).