

МАТЕМАТИКА ДЛЯ ПРОГРАММИСТА

Одна из первых проблем, с которой сталкивается студент-первокурсник, – это различие требований, предъявляемых к нему при изучении математики в школе и в университете. Речь идёт не о «больших» или «меньших» требованиях, а о принципиально разных подходах к оценке «достижений» студента.

Хотя школьная программа построена таким образом, чтобы за время обучения ученик получил некие фундаментальные математические знания, владение которыми поможет успешно справиться и с университетской программой, на деле школа большинством даже хороших учеников воспринимается как некий трамплин, который позволит успешно сдать централизованное тестирование и поступить в вуз. Поэтому основное внимание школьники уделяют не овладению базовыми математическими понятиями и инструментами, не попыткам понять логику предмета, не умению строить строгие логические цепочки, доказывать и обосновывать утверждения, а приобретению навыков решения конкретных задач определённых типов. При изучении очередной темы первый вопрос, который волнует ученика, – это не «как это можно использовать», а «будет ли эта тема в ЦТ», и, если не будет, то и изучать её незачем. Безусловно, навыки решения задач весьма полезны и необходимы и, конечно, высокий балл на централизованном тестировании значительно расширяет возможности для выбора вуза и специальности. Но такой подход приводит к тому, что студенты даже такой популярной и востребованной специальности, как «Информационные системы и технологии», имеющие один из самых высоких баллов по ЦТ в университете (в том числе, и по математике), совершенно серьёзно полагают, что программист – это человек, который пишет программы, используя языки программирования и стандартную библиотеку алгоритмов, а математика к его работе не имеет никакого отношения.

Всё это формирует у большинства потенциальных студентов представление о математике как о науке, главной задачей которой является решение неких примеров и задач по определённым схемам, и мнение о том, что студенту достаточно отработать навыки решения нескольких классов типовых задач, чтобы утверждать, что он знает математику.

Однако уже на первом курсе вчерашние ученики оказываются в растерянности: при изучении университетского курса математики студент должен не только и не столько натренироваться в решении

типовых задач, сколько ознакомиться с различными математическими теориями, понять их и уметь применить на практике и, кроме того, на основе полученного ответа сделать некий вывод. Вполне естественно, что непривычные требования не вызывают особого энтузиазма: лекционные занятия большинством студентов воспринимаются как непонятный «довесок» к «главному» – практическим занятиям, на которых можно делать привычные действия – считать по формулам. При этом любые попытки придать математическим вычислениям реальный смысл вызывают сопротивление: студент легко вычислит производную, но далеко не каждый найдет мгновенную скорость в точке, при вычислении интеграла от неотрицательной функции получит в ответе, например, ноль, но будет абсолютно уверен, что всё сделал правильно, т. к. в его сознании вычисление интеграла (практический навык) никак не связан с площадью (геометрический смысл определённого интеграла).

Часто студент бывает разочарован и своей оценкой на экзамене: умение решать типовые задачи в школе гарантировало, по крайней мере, не ниже восьми баллов. В вузе же решение без каких-либо обоснований и объяснений, почему нужно делать именно так и что из этого следует, оценивается гораздо ниже. Цель изучения математики в университете заключается не просто в выработке умений и навыков решения задач, а в формировании привычки аналитического и логического мышления. В итоге, задача университета – подготовить специалиста, который сможет самостоятельно анализировать и решать поставленные перед ним задачи. Возможно, в будущей профессии программистам, например, не пригодятся производные или матрицы, но если и пригодятся, то именно в прикладных задачах, предполагающих знание общей теории, а не в виде типового примера. Конечно, вряд ли современный IT-специалист будет вручную строить графики функций, но умение собирать, анализировать и интерпретировать информацию ни для кого не будет лишним. Математика пригодится тем, кто занимается анализом больших данных (Big Data), машинным обучением, причём совершенно разные её аспекты, нельзя даже представить заранее, какая именно область математики им может понадобиться. Конечно же, программисту необходимо знать алгоритмы и статистику. Какую бы программу он ни писал, ему нужно будет оценить её сложность и производительность. Причем оценить заранее, до написания кода, чтобы понять, оптимальный ли алгоритм выбран для её реализации. Закладывать красивые решения в самом начале проектирования системы – это основа основ. А для того чтобы научиться видеть это красивое решение, необходимо иметь базовые фундаментальные знания.