Вопрос No1   
Да, я читал чужой код. Честно говоря у меня небольшой рабочий опыт был. Мне приходилось разбираться в плохо спроектированном коде, состоящем из одной большой процедуры, в который ежедневно вносили изменения несколько разработчиков, увы не помню что была за процедура, но впечатления были не ахти какие... К тому же я увидел большое количество «костылей» ... вследствие чего начинаю отчетливо понимать, насколько важно писать чистый код.

Вопрос No2   
Перечислю языки о которых я читал и помню:   
**C**​ – прародитель С++ и С#, язык «низкоуровневого программирования» на котором разрабатывают, например, компоненты операционной системы. Является стандартизированным процедурным ЯП.  
**C#**​ - гибкий, универсальный язык для всех направлений (консольные программы, и программы с графическим интерфейсом. Поддерживается компанией Microsoft. Разработан для платформы .NET Framework . Объектно-ориентированный язык программирования.  
**SQl**​ – ​язык структурированных запросов. Не процедурный, создан для управления БД для реляционных баз данных. Имеет некие расширения, который увеличивают его функционал до полного языка программирования.   
**PERL**​ – интерпретируемый язык программирования, созданный для работы с регулярными выражениями. Используется в web разработке, играх...   
**PHP**​ – ЯП с открытым исходным кодом, разработанный для написания сайтов и веб-разработок( быстрого создания web страниц).  
**JAVA**​ – Объектно-ориентированный ЯП. Создан для написания игр и приложений на Android, Знаю что программисты с опытом работы на этом языке очень востребованы.  
**JAVASCRIPT**​ - используется для написания Front-end части, также для серверного программирования.

Вопрос No3   
Компилятор – это программа, которая позволяет компилировать написанное - переводить в машинные команды. А у не компилируемых языков должна быть своя программа – интерпретатор, которая исполняет текст программы шаг за шагом, не переводя его в язык машины.​ ​Да, компилируемые языки производительнее, но у них нет динамической типизации. Писать на них сложнее, их нужно перекомпилировать под разные платформы (обычно кроме языков, компилирующихся в байт-код, как java и C#) Есть интересный момент - c# и java являются одновременно и компилируемыми и интерпретируемыми. Они сначала компилятся в байт-код, а потом этот байт-код интерпретируется. А интерпретируемые программы можно просто запускать на любой платформе при наличии интерпретатора. У интерпретируемый языков есть возможность легко изменять код, без каких-либо действий (компиляции).

Вопрос No4   
Это рабочая область, которая упрощает работу разработчику в создании программного продукта: web – сайтов и web – сервисов, мобильных и десктопных приложений. В принципе, если опираться на фреймворк, то не нужно изобретать колесо заново. Нужно только выбрать подходящую архитектуру и при желании остаётся только дополнить её своим кодом. Да, это всё стабильно работает, фреймворк учитывает огромное количество нюансов, однако теряется гибкость. Работал с ​Bootstrap.

Вопрос No5 HTTP(​*HyperText Transfer Protocol*​) – протокол передачи гипертекста, данных в сети. Внутри прописаны некие “действия”, с помощью которых браузер загружает web-страницы, а сервер получает данные, которые ввёл пользователь на сайте. HTTPS (​*HyperText Transfer Protocol Secure*​) – это расширение протокола HTTP, способствующее конфиденциальности информации путём её шифрования. Используется, например, на сайтах с авторизацией, взаимодействиях с платежными системами, почтовыми сервисами. Шифровка данных позволяет исключить возможность их получения и использование третьим лицам.

Вопрос No6 1) Начнем с Toolbar - это синия панель с иконкой поиском и разными кнопками, в ней есть внутри компонент Avatar это title + img. 2) далее будет sidebar это панель со списком "моя страница и тд можно его как navbar назвать 3) Далее PagePhoto - это где фотка и пара кнопок. 4)PageInfo или PageSummary это краткая информация с датой рождения extentinPageInfo это компонент при нажатии "показать подробную информацию" 5) Дальше PageTabs это список табов (Это где записи Павла и синия линия внизу) 6) Дальше компонент Post, пост с фотками лайками и тд, 7) Комментарии.

Вопрос No7 SQL это универсальный язык запросов к реляционным базам данных. Почему многие СУБД его используют? 1) он достаточно мощный 2) Чтобы понизить уровень вхождения новых пользователей Дело в том, что sql это язык запросов, предназначенный для реляционных баз, в то время, как, например, mongo - документ-ориентированная. SQL ещё так популярен, потому что это декларативный язык, что позволяет использовать его в любой реляционной СУБД, независимо от её устройства.

Вопрос No8 ссылка: ​https://github.com/triplex239/MathDLL

Вопрос No9 https://github.com/triplex239/MathDLL/blob/master/MyMath.cs

Комментарии в конце кода.