

# Практические задания

Это очень важная часть обучения. Если вы их не делаете или делаете поверхностно, то отдаляете дату вашего первого оффера (предложения работодателя). Вам придётся проходить собеседования, и только колоссальный объём работы по написанию кода поможет достойно отвечать на вопросы.

## Советы и правила:

- 1. Не откладывайте выполнение домашнего задания** на следующий урок. Как правило, это приводит к серьёзному отставанию в понимании материала.
- 2. Воспринимайте оценку преподавателя как показатель вашего прогресса.** Если вам поставили 3 или «не сдано» — это сигнал, что материал не усвоен, и нужно приложить двойные усилия. Задача преподавателя — показать реальный прогресс, а не его иллюзию. Представьте, что это ваш работодатель. Его оценка — метрика, насколько вы интересны как потенциальный сотрудник.
- 3. Очень важно внимательно несколько раз перечитать формулировку задачи перед её решением.** На собеседованиях очень часто проверяют навык точного понимания задачи. Если вам что-то непонятно, надо задать вопрос преподавателю, наставнику, одногруппникам или в чате вашей группы в Telegram.
- 4. Если не сказано иного, при решении задач нельзя использовать внешние библиотеки и модули.** Сейчас главная задача — реализовать алгоритм своими силами, а не путём импорта готового решения.
- 5. Думайте на шаг вперёд:** из-за чего может сломаться код? Какие данные могут быть недопустимы? Что будет, если масштаб задачи увеличится на порядок или на несколько порядков? Помните, что если в домашнем задании был список, состоящий из 10 элементов, то в реальных условиях размер списка может быть 10 000, 100 000 и больше. Это должно предостеречь вас от некоторых заведомо неэффективных способов решения задачи.
- 6. Всегда проверяйте код перед отправкой домашнего задания.** Прислать нерабочий код — очень грубая ошибка.

## Как выполнять домашние задания и сдавать их на проверку?

Создайте репозиторий на GitHub, в котором будут храниться решения домашних заданий. Не забудьте добавить в репозиторий соответствующий файл [.gitignore для Visual Studio](#). Внимательно следите за содержимым коммитов. Убедитесь, что в коммит не попадают бинарные или временные файлы, а также давайте коммитам осмысленные комментарии.

В корне репозитория должны быть отдельные папки под каждый урок. Если домашнее задание содержит несколько задач, необходимо создать вложенные папки с номерами задач.

Перед решением задач необходимо создать ветку с номером урока (например, lesson-1), вести в ней всю разработку, а в конце создать pull-request в ветку main своего репозитория. Предоставить преподавателю ссылку на pull-request.

Рекомендуем создавать к каждому домашнему заданию файл readme.txt, с кратким изложением итогов выполнения дз по пунктам (выполнено, сложно, легко). Если что-то не получилось — напишите, что сделали, сколько времени ушло на попытки решить задачу. Это хорошая обратная связь для преподавателя и повод что-то дополнительно рассказать в начале урока. А вам это поможет следить за собственным прогрессом и оценивать пробелы в знаниях.

Домашнее задание желательно отправлять как можно раньше, чтобы преподаватель успел проверить все работы до начала нового урока. Необходимо предоставить решение в крайнем случае до вечера перед следующим уроком. Официально домашнее задание можно сдать за 4 часа до начала урока, но не все преподаватели успевают проверить за такое короткое время.

**Обязательно** закладывайте запас в несколько часов на отладку возможных багов и поиск ошибок в коде. Были случаи, когда поиск одной неправильной запятой занимал несколько часов. Программирование требует большой концентрации внимания, и этот навык только вы сами можете выработать во время поиска опечаток в коде.

**Совет!** Никогда не копируйте чужой код (в том числе и наши исходники с урока), вместо этого располагайте его в соседнем окне и печатайте **своими** руками. Это важно для наработки навыка.