1. Разработайте компонент, позволяющие динамически компилировать и выполнять скомпилированный код, написанный на C#.

Предполагается, что этот компонент будет использоваться в системе проверки знаний, в которой студент будет писать некоторый код (метод или класс), который проверяющая система должна будет скомпилировать, а затем вызвать заданные методы.

**Замечания по реализации**:

* Написанный код может представлять из себя отдельный метод, для работы которого может потребоваться какое-то окружение (другие методы, переменные, константы). Поэтому нужно учитывать не только тот код, который будет писать проверяемый, но и заготовку кода, которую будет предоставлять потенциальный составитель задания.
* В процессе не должно создаваться исполняемых файлов, результатом работы компонента должен быть некий объект, для которого можно будет вызвать заданный метод.
* Следует учитывать, что компилируемые методы могут иметь разное число параметров, следует учитывать это при предоставлении api для вызова этого метода.
* Компонент должен работать в среде .NET не ниже 5й версии. Для решения задачи можно использовать класс CSharpCompilation (+почитать про рефлексию/отражения в C#, Emit API).

1. Настройка CI/CD конвеера, автоматическая сборка, деплой приложения. Составить отчет, в котором описать основные шаги.
2. Реализация компонента, поддерживающего динамические биндинги через CustomTypeDescriptor.
3. Доработка генератора классов.
4. Доработка телеграмм-бота расписания СибУПК, один из вариантов:
   1. Добавить команды для повторной инициализации базы данных, хранящей информацию о преподавателях и расписании. Сейчас это выполняется вручную – служба бота останавливается, затем удаляется база данных и после рестарта данные загружаются заново. Команда должна быть доступна только для админской учетки и ее выполнение не должно приводить к нарушениям работы – загрузка должна происходить в фоне, и только после окончания инициализации должна происходить подмена источника данных.
   2. Изменить механизм рассылки уведомлений об обновлениях расписания: рассылать их только тем, кто делал запросы в течение последних 2 недель.
   3. Реализовать очистку лога действий пользователей старше 2 месяцев, либо архивирование.
   4. Рассылать уведомления об обновлении расписания только тем, у кого в данный момент выбрана группа, и только в том случае, если есть изменения для данной группы.
   5. Реализовать кэширование расписания для преподавателей.
5. Разобраться с реактивным фреймворком. На его базе создать заготовку для клиент-серверного приложения (на примере чата с широковещательной рассылкой). Окончательный вариант задания согласуется с преподавателем.