**Методы проектирования, используемые при создании проекта:**

**Метод декомпозиции.**

Данный метод подразумевает, что любой объект будет рассматриваться как сложную систему, состоящую из нескольких подсистем, которые в свою очередь также могут быть поделены на отдельные части.

Конкретно в данном проекте декомпозиция будет использоваться для разработки отдельных частей одной системы. Это влечёт собой следующие преимущества:

* Уменьшается время, занимаемое на разработку:

Это позволяет разбить исходный код программы на отдельные модули, которые могут разрабатываться несколькими разработчиками параллельно, причём реализация одного модуля не будет зависеть от реализации другого модуля. Так же готовые модули одной программы можно использовать при разработке другой.

* Повышается надёжность программы:

В случае возникновения ошибки, разработчикам не приходится тратить большое количество времени на поиск места возникшей ошибки, она будет локализована в том модуле, в котором она возникла. По этой же причине отладка программы, которая состоит из отдельных модулей, становится менее трудоёмкой.

В данном проекте этот метод будет применяться при разработке программы. Программа будет иметь графический интерфейс, который будет реагировать на действия пользователя посредством вызова выполнения определённых команд. Например, если пользователь нажмёт на кнопку вывода списка своих одногруппников, то произойдёт вызов метода, который будет получать список одногруппников для текущего пользователя. После чего список будет выведен в интерфейс программы в удобном для восприятия виде.

**Метод итерации.**

Данная методика решает проблемы неточностей и противоречий исходных данных в процессе проектирования.

В случае долгой разработки приложения, некоторые из модулей могут некорректно работать с другими модулями, а также могут появляться дополнительные условия работы ранее проектируемых модулей.

Методика решения данных проблем состоит в выполнении итерационных процедур. Первый цикл итерации решения поставленной задачи состоит в взятии в расчёт ограниченное число исходных данных и учитываемых факторов. С последующими циклами итераций мы возвращаемся в начало поставленной задачи и повторяем её решение, но уже с уточнёнными значениями исходных данных и т.д. до тех пор, пока поставленная задача не будет выполнена.

Данный метод разработки будет применяться в процессе реструктуризации исходного кода программы (далее рефакторинг). После реализации нового функционала программы, проводится исследование структуры исходного кода на дублирование функций, временные решения, а также делается тестирование программы и поиск возникших ошибок. После чего принимается решение, стоит ли проводить рефакторинг или нет. В процессе рефакторинга методы могут быть переписаны заново, с учётом нового функционала. Это позволяет:

* Упростить дальнейшую разработку и сопровождение программы: по возможности новые методы должны использовать общие функции и классы программы.
* Упросить тестирование и отладку программы: в случае возникновения ошибки в работе программы легче понять, где она возникла, и одновременно с этим она будет локализована в том программном модуле, в котором она возникла.