Сказка

212 Завьялова Милана

30.09.2023

Здравствуйте, Михаил Иванович!

Сегодняшнее занятие было записано вами для нас и выложено на YouTube. В видеороликах были просмотрены папки с домашними заданиями (сказки, эссе и код). Были замечания по оформлению сказки (писать сверху число) и о стиле написания (стоит более эмоционально подходить к выполнению данного домашнего задания).

Далее мы начали разбирать нашу дальнейшую работу по проекту “Ровер на бездорожье”.

Зачем разделять boundary и control? Boundary на клиенте (одно устройство), control на сервере (другое устройство), общаются по сети (вопрос следующего семестра). Работа программы: main читает конфигурационный файл (в нем указаны по умолчанию некоторые параметры (имя файла с boundary, где будут писаться логи), кроме этого control пишет логи (после приема команды от boundary, вызывает необходимую генерацию, далее пишет лог что команда отработана, а boundary пишет что код ошибки 0). По сути это некоторая беседа-отчетность о выполненных командах, кодах ошибки в виде текстовых файлов.Кроме этого в конфигурационном файле пишутся имена двух текстовых файлов для логов работы boundary на клиенте и работы control на сервере, пишется время (есть/нет) для каждого лога (отправки и получения ответа для boundary, приеме и отправки ответа для control), есть/нет дата, параметры по умолчанию для генерации (дисперсия 1, мин среднее 1), *текущее расположение курсора генерации* (по умолчанию - центр поля), команды передвижения курсора на Δx и Δy или на абсолютные координаты. В конфигурационном файле написан командный файл(может быть не один, и пишем кол-во и их имена) для выполнения (на отладке можем делать с клавиатуры).

Main читает конфигурационный файл, вызывает интерфейс. Интерфейс начинает читать команды из командного файла, распознавать их, если все хорошо, то отправлять команду control на выполнение. Control должен иметь все команды, которые может выполнять.

Дз (7-14): реализация boundary и control и выше описанный процесс. И их диаграмма классов.