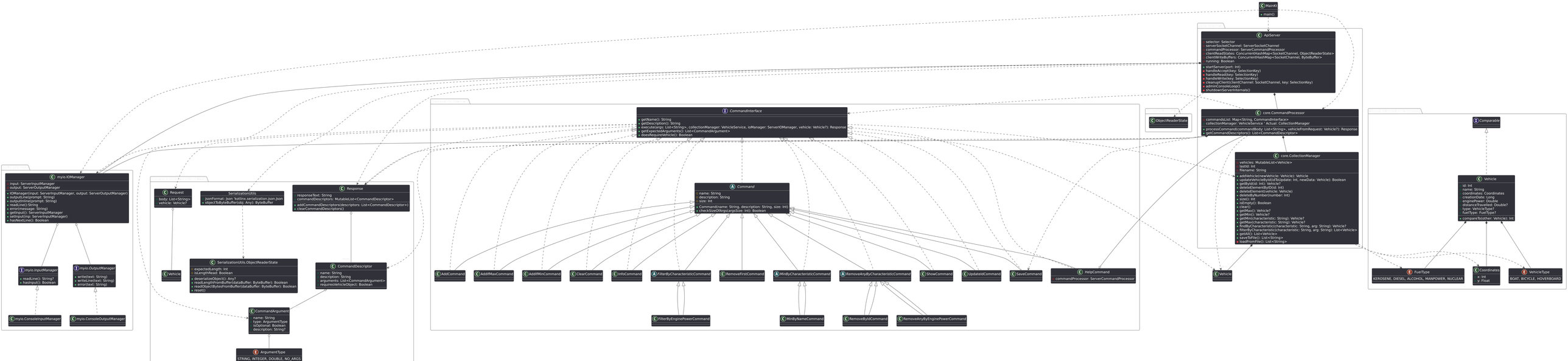
MainKt
+ main()
ApiServer
- selector: Selector
- serverSocketChannel: ServerSocketChannel
- commandProcessor: ServerCommandProcessor
- clientReadStates: ConcurrentHashMap<SocketChannel, ObjectReaderState>
- clientWriteBuffers: ConcurrentHashMap<SocketChannel, ByteBuffer>
- running: Boolean
+ startServer(port: Int)
# handleAccept(key: SelectionKey)
# handleRead(key: SelectionKey)
# handleWrite(key: SelectionKey)
# cleanupClient(clientChannel: SocketChannel, key: SelectionKey)
# adminConsoleLoop()
# shutdownServerInternals()
ObjectReaderState
core.CommandProcessor
- commandsList: Map<String, CommandInterface>
- collectionManager: VehicleService ^Actual: CollectionManager
+ processCommand(commandBody: List<String>, vehicleFromRequest: Vehicle?): Response
+ getCommandDescriptors(): List<CommandDescriptor>
core.CollectionManager
- vehicles: MutableList<Vehicle>
- initid: Int
- filename: String
+ addVehicle(newVehicle: Vehicle): Vehicle
+ updateVehicleById(idToUpdate: Int, newData: Vehicle): Boolean
+ getById(id: Int): Vehicle?
+ deleteById(id: Int): Boolean
+ deleteElement(vehicle: Vehicle)
+ deleteByNumber(number: Int)
+ size(): Int
+ isEmpty(): Boolean
+ clear()
+ getMax(): Vehicle?
+ getMin(): Vehicle?
+ getMin(characteristic: String): Vehicle?
+ getMax(characteristic: String): Vehicle?
+ findByCharacteristic(characteristic: String, arg: String): Vehicle?
+ removeAnyByCharacteristic(characteristic: String, arg: String): List<Vehicle>
+ getAll(): List<Vehicle>
+ saveToFile(): List<String>
+ loadFromFile(): List<String>
Comparable
Vehicle
- id: Int
- name: String
- coordinates: Coordinates
- creationDate: Long
- enginePower: Double
- distanceTraveled: Double?
- fuelType: FuelType?
- vehicleType: VehicleType?
+ compareToother: Vehicle): Int
FuelType
KEROSENE, DIESEL, ALCOHOL, MANPOWER, NUCLEAR
Coordinates
- x: Int
- y: Float
VehicleType
BOAT, BICYCLE, HOVERBOARD
CommandInterface
+ getName(): String
+ getDescription(): String
+ execute(arge: List<String>, collectionManager: VehicleService, ioManager: ServerIOManager, vehicle: Vehicle?): Response
+ getExpectedArguments(): List<CommandArgument>
+ doesRequireVehicle(): Boolean
Command
- name: String
- description: String
- size: Int
+ Command(name: String, description: String, size: Int)
+ checkSizeOfArgs(argsSize: Int): Boolean
AddCommand
AddIfMaxCommand
AddIfMinCommand
ClearCommand
InfoCommand
FilterByCharacteristicCommand
RemoveFirstCommand
MinByCharacteristicCommand
RemoveAnyByCharacteristicCommand
ShowCommand
UpdateIdCommand
SaveCommand
HelpCommand
- commandProcessor: ServerCommandProcessor
FilterByEnginePowerCommand
MinByNameCommand
RemoveByIdCommand
RemoveAnyByEnginePowerCommand
myio.IOManager
- input: ServerInputManager
- output: ServerOutputManager
+ IOManager(input: ServerInputManager, output: ServerOutputManager)
+ outputLine(prompt: String)
+ outputInline(prompt: String)
+ readLine(): String
+ readInline(): String
+ getInputs(): ServerInputManager
+ setInputs(ip: ServerInputManager)
+ hasNextLine(): Boolean
myio.InputManager
+ readLine(): String?
+ hasInput(): Boolean
myio.OutputManager
+ write(text: String)
+ writeLn(text: String)
+ error(text: String)
myio.ConsoleInputManager
myio.ConsoleOutputManager
Request
- body: List<String>
- vehicle: Vehicle?
SerializationUtils
- jsonFormat: Json 'kotlinx.serialization.json.Json'
+ objectToByteBuffer(obj: Any): ByteBuffer
Response
- responseText: String
- commandDescriptors: MutableList<CommandDescriptor>
+ addCommandDescriptors(descriptors: List<CommandDescriptor>)
+ clearCommandDescriptors()
Vehicle
SerializationUtils.ObjectReaderState
- expectedLength: Int
- isLengthRead: Boolean
+ deserializeObject(): Any?
+ readLengthFromBuffer(dataBuffer: ByteBuffer): Boolean
+ readObjectBytesFromBuffer(dataBuffer: ByteBuffer): Boolean
+ reset()
CommandDescriptor
- name: String
- description: String
- arguments: List<CommandArgument>
- requiresVehicleObject: Boolean
CommandArgument
- name: String
- type: ArgumentType
- isOptional: Boolean
- description: String?
ArgumentType
STRING, INTEGER, DOUBLE, NO_ARGS