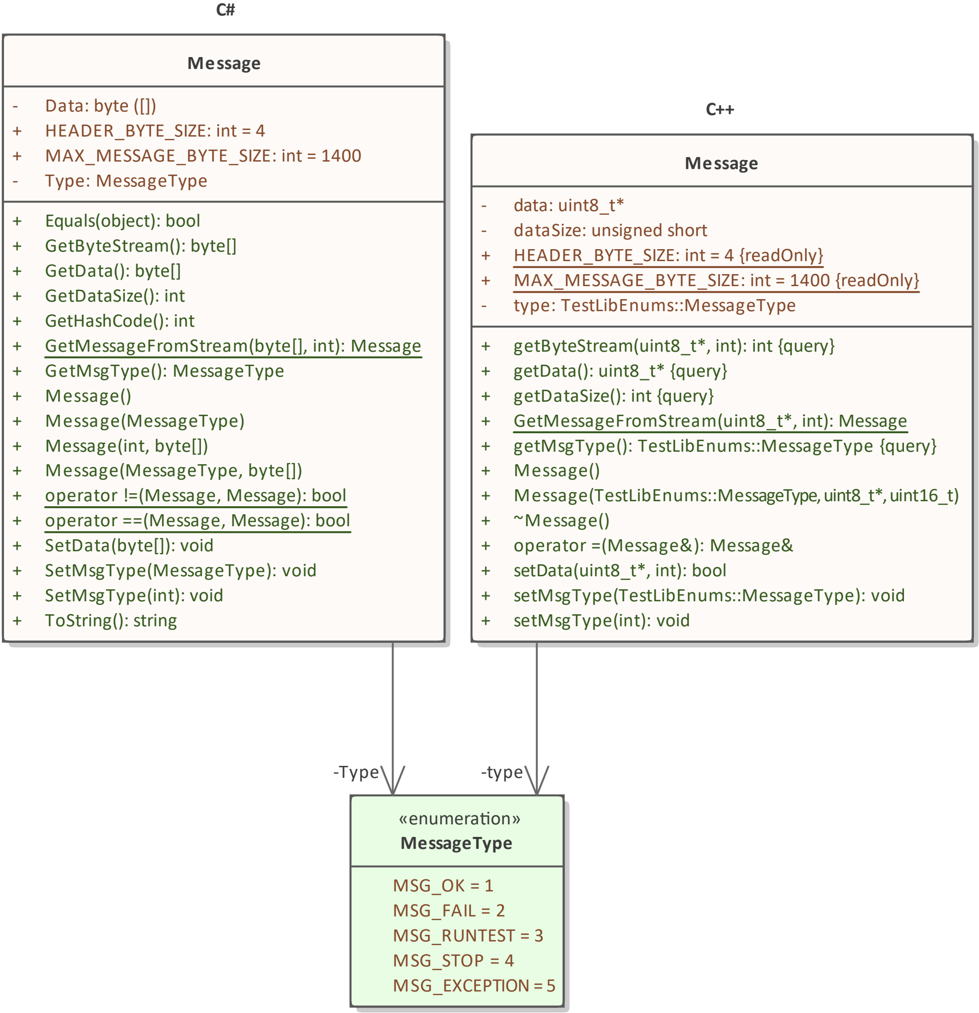
C#
Message
- Data: byte ([])
+ HEADER_BYTE_SIZE: int = 4
+ MAX_MESSAGE_BYTE_SIZE: int = 1400
- Type: MessageType
+ Equals(object): bool
+ GetByteStream(): byte[]
+ GetData(): byte[]
+ GetDataSize(): int
+ GetHashCode(): int
+ GetMessageFromStream(byte[], int): Message
+ GetMsgType(): MessageType
+ Message()
+ Message(MessageType)
+ Message(int, byte[])
+ Message(MessageType, byte[])
+ operator !=(Message, Message): bool
+ operator ==(Message, Message): bool
+ SetData(byte[]): void
+ SetMsgType(MessageType): void
+ SetMsgType(int): void
+ ToString(): string
C++
Message
- data: uint8_t*
- dataSize: unsigned short
+ HEADER_BYTE_SIZE: int = 4 {readOnly}
+ MAX_MESSAGE_BYTE_SIZE: int = 1400 {readOnly}
- type: TestLibEnums::MessageType
+ getByteStream(uint8_t*, int): int {query}
+ getData(): uint8_t* {query}
+ getDataSize(): int {query}
+ GetMessageFromStream(uint8_t*, int): Message
+ getMsgType(): TestLibEnums::MessageType {query}
+ Message()
+ Message(TestLibEnums::MessageType, uint8_t*, uint16_t)
+ ~Message()
+ operator =(Message&): Message&
+ setData(uint8_t*, int): bool
+ setMsgType(TestLibEnums::MessageType): void
+ setMsgType(int): void
-Type
-type
«enumeration»
MessageType
MSG_OK = 1
MSG_FAIL = 2
MSG_RUNTEST = 3
MSG_STOP = 4
MSG_EXCEPTION = 5