ExchangeRateService

- remoteRepository: IRemoteRepository
- encoder: IEncoder
- documentParser: IDocumentParser
- cachedTable: ExchangeTable
- NBP_URL: String {static, final}

+ ExchangeRateService(remoteRepository: IRemoteRepository, encoder: IEncoder, documentParser: IDocumentParser) throws Exception
- fetchAndParseTableInternal(): ExchangeTable throws Exception
+ fetchAndParseTable(): ExchangeTable throws Exception
+ clearCache(): void

<<USE>>

<<USE>>

<<Interface>>
IDocumentParser

+ getTable(rawData: String): ExchangeTable

<<Interface>>
IEncoder

+ encodeBytes(bytes: byte[]): String

DocumentParser

+ getTable(rawData: String): ExchangeTable throws Exception
- safeParseInt(s: String): int
- safeParseDouble(s: String): double

Encoder

- NBP_CHARSET: Charset {static, final}

+ encodeBytes(bytes: byte[]): String

<<CREATE>>

<<CREATE>>

Main

+ main(args: String[]): void

<<USE>>

<<Interface>>
IRemoteRepository

+ getUrlBytes(url: String): byte[]

RemoteUrlRepository

+ getUrlBytes(urlString: String): byte[]
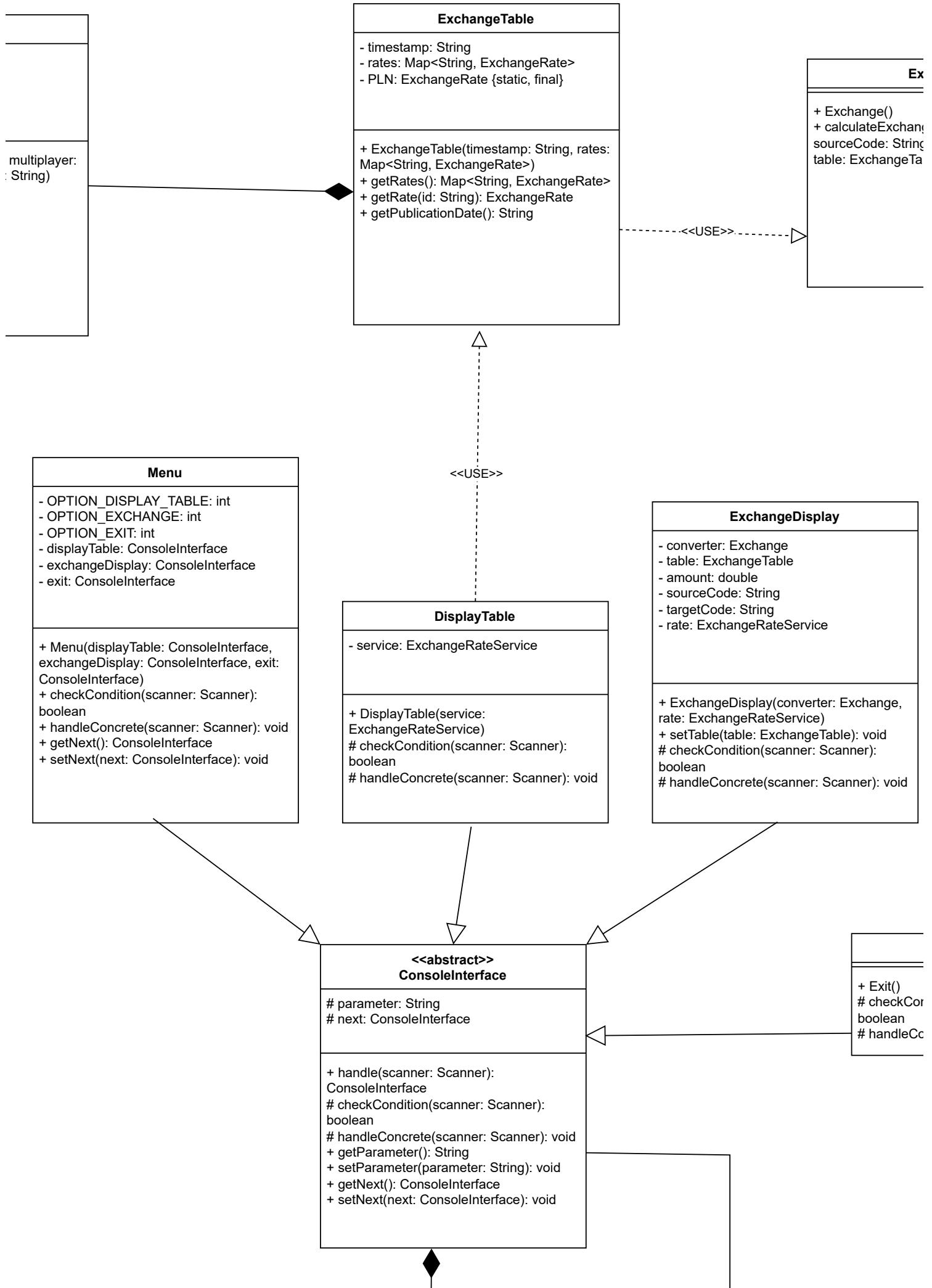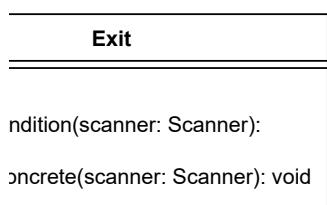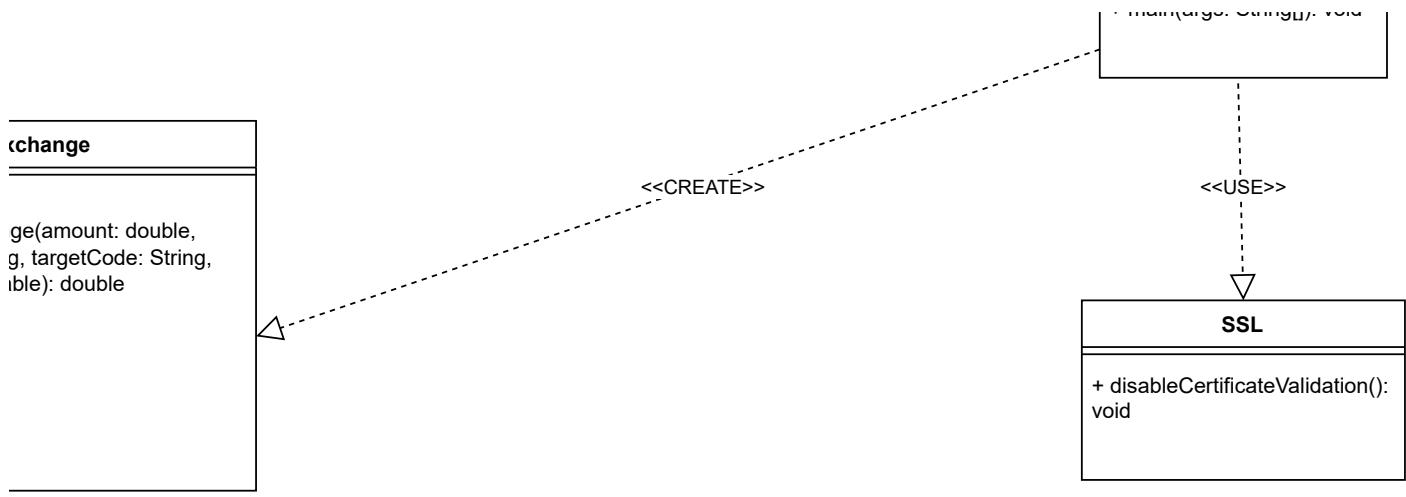throws Exception

<<CREATE>>

ExchangeRate

- rate: double
- multiplayer: double
- name: String
- idString: String

+ ExchangeRate(rate: double,
double, name: String, idString:
+ getRate(): double
+ getMultiplayer(): double
+ getName(): String
+ getIdString(): String
+ getUnitRate(): double
+ equals(o: Object): boolean
+ hashCode(): int

ExchangeTable

- timestamp: String
- rates: Map<String, ExchangeRate>
- PLN: ExchangeRate {static, final}

+ ExchangeTable(timestamp: String, rates: Map<String, ExchangeRate>)
+ getRates(): Map<String, ExchangeRate>
+ getRate(id: String): ExchangeRate
+ getPublicationDate(): String

<<USE>>

Ex...

+ Exchange()
+ calculateExchang...
sourceCode: String
table: ExchangeTa...

Menu

- OPTION_DISPLAY_TABLE: int
- OPTION_EXCHANGE: int
- OPTION_EXIT: int
- displayTable: ConsoleInterface
- exchangeDisplay: ConsoleInterface
- exit: ConsoleInterface

+ Menu(displayTable: ConsoleInterface, exchangeDisplay: ConsoleInterface, exit: ConsoleInterface)
+ checkCondition(scanner: Scanner): boolean
+ handleConcrete(scanner: Scanner): void
+ getNext(): ConsoleInterface
+ setNext(next: ConsoleInterface): void

<<USE>>

DisplayTable

- service: ExchangeRateService

+ DisplayTable(service: ExchangeRateService)
# checkCondition(scanner: Scanner): boolean
# handleConcrete(scanner: Scanner): void

ExchangeDisplay

- converter: Exchange
- table: ExchangeTable
- amount: double
- sourceCode: String
- targetCode: String
- rate: ExchangeRateService

+ ExchangeDisplay(converter: Exchange, rate: ExchangeRateService)
+ setTable(table: ExchangeTable): void
# checkCondition(scanner: Scanner): boolean
# handleConcrete(scanner: Scanner): void

<>
ConsoleInterface

parameter: String
next: ConsoleInterface

+ handle(scanner: Scanner): ConsoleInterface
# checkCondition(scanner: Scanner): boolean
# handleConcrete(scanner: Scanner): void
+ getParameter(): String
+ setParameter(parameter: String): void
+ getNext(): ConsoleInterface
+ setNext(next: ConsoleInterface): void

+ Exit()
# checkCon...
boolean
handleCo...

multiplayer:
String)

xchange

ge(amount: double,
g, targetCode: String,
ble): double

+ main(args: String[]): void

<<CREATE>>

<<USE>>

SSL

+ disableCertificateValidation():
void

Exit

ndition(scanner: Scanner):

oncrete(scanner: Scanner): void