

# Automatický obchodní systém

Jan Kvapil

# Úvod

- Cílem vytvořit automatický obchodní systém
- Vytvořená aplikace komunikuje s obchodní platformou MetaTrader, ze které získává obchodní data
- Automaticky realizuje obchodní pokyny na základě uživatelsky definovaných obchodních strategií
- Hlavní předností aplikace je jednoduchá definovatelnost obchodních strategií v porovnání s nástroji dostupnými v rámci platformy MetaTrader

# Obsah

- Základní pojmy
- Platforma MetaTrader
- Architektura aplikace
- Použité technologie
- Popis aplikace

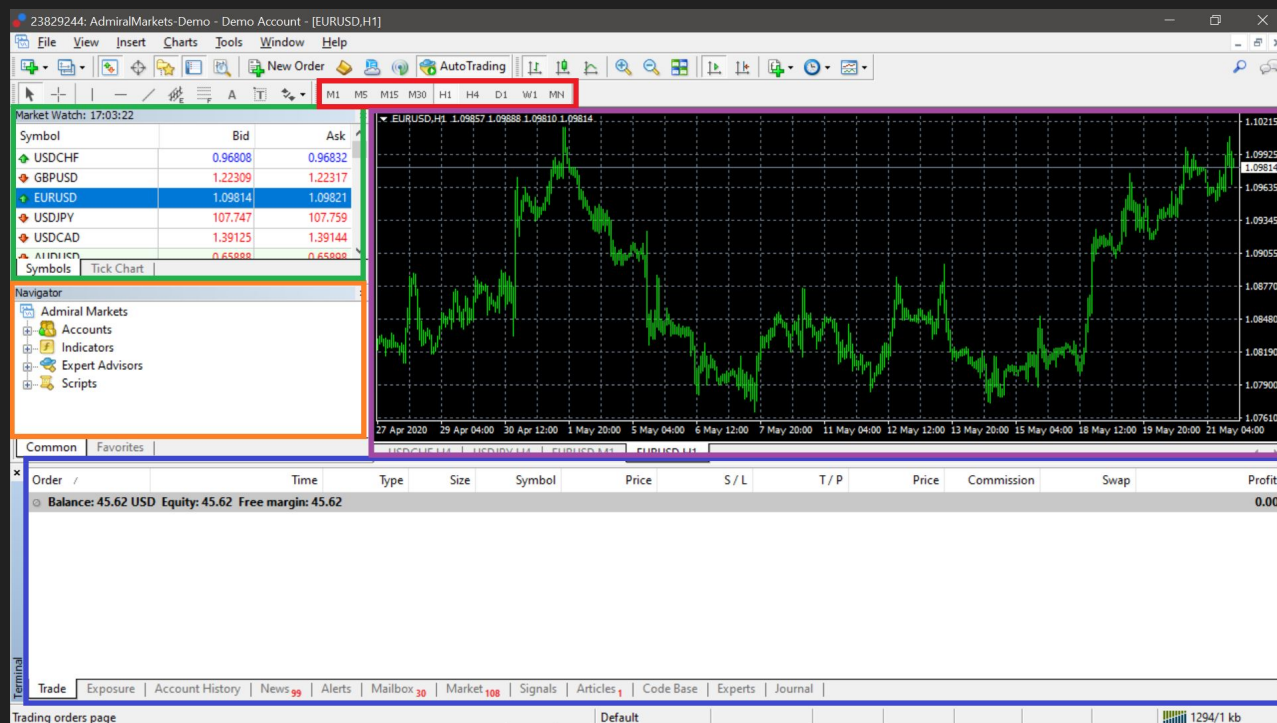
# Základní pojmy

- *Instrument* = investiční nástroj (akcie, kryptoměny, měnové páry, ...)
- *Burza* = instituce realizující trh s instrumenty
- *Forex* = typ burzy (směna měnových párů)
- *Broker* = společnost poskytující přístup k burze individuálním obchodníkům

# Platforma MetaTrader

- Poskytuje UI pro obchodování
- Klient se přihlásí skrze tuto platformu k obchodnímu účtu u brokera (demo/real)
- Programovatelná (MQL)

# Platforma MetaTrader



# Základní pojmy

- Cena (bid/ask)
- Pohyb ceny
- Svíčkový graf
- Časové období
- Obchodní pokyn
- SL/TP



# Základní pojmy

- Technická analýza trhu (vývoj ceny)
- Indikátory - statistické metody vycházející z předešlého vývoje ceny
- Základní typy indikátorů implementovány přímo v MetaTraderu
- Moving Average



# Moving Average



# Obchodní strategie

- Strategie vycházející z tech. analýzy
- Určují kdy odeslat obchodní pokyny  
(spekulace nad zvýšením/poklesem hodnoty instrumentu s ohledem na riziko)
- Automatizovatelné = AOS

# Architektura aplikace

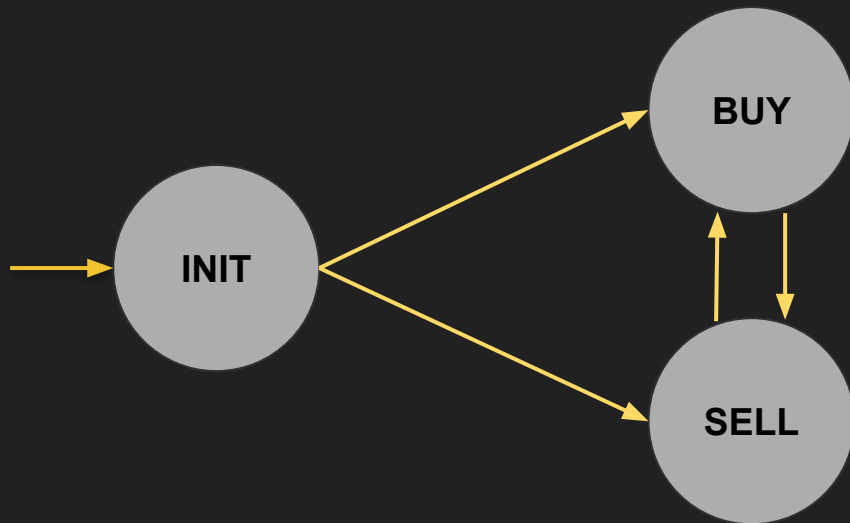
- Rozdělena na klient (js) / server (mql)
- Server: vykonává obchodní pokyny (DWX)
- Komunikace pomocí zpráv

"TRADE | ACTION | TYPE | SYMBOL | PRICE | SL | TP | COMMENT | TICKET"

- Klient: GUI pro připojení k serveru a tvorbu strategií + automatická správa strategií

# Architektura aplikace

- Strategie jako stavový automat



# Architektura aplikace

- Definice přechodových funkcí

```
// SELL predicate
```

```
price < indicators.get("ma100")
```

```
// BUY predicate
```

```
price >= indicators.get("ma100")
```

# Architektura aplikace

- Definice vlastních indikátorů

```
exports.average = n => {  
  return db => {  
    if (db.length !== n)  
      throw new Error(`Wrong size of array.`)  
    return db.reduce((a, b) => a + b, 0) / n  
  }  
}
```

# Použité technologie

- Electron (node.js), React, ZeroMQ

```
// React component
const Button = (props) => {
  return (
    <button onClick={props.onClick}>
      { props.value }
    </button>
  )
}
```

# Popis aplikace

- [github](#) (video)