

주식자동매매 프로그램

- 순서

1. 개요
2. 기술
3. 프로그램 기능소개
4. 주기능 Source
5. 끝맺음

- 1. 개요

- 키움증권 Open Api를 이용한 주식자동매매 프로그램으로 사용자가 설정 한 조건식을 바탕으로 자동매매가 이루어집니다. 자동매매 프로그램을 활용함으로써 빠르게 반응하여 사용자가 세팅한 조건식의 정확성을 높일 수 있습니다.
- 사용자 조건식을 바탕으로 손절, 익절 포인트를 미리 설정해 해당 포인트에 도달하면 설정된 전략에 따라 자동매매를 진행합니다.

- 2. 기술

- C#
- 키움증권 Open Api
- WinForm
- VisualStudio2019

- 3. 기능소개 -전체 화면

컨트롤창

Form1

로그인중입니다.

계좌번호

8024053711

예수금

D+2주정예수금(예약자산평가액)

당일투자원금

당일투자손익

당일손익률

210,662,273원

211,523,784원

0원

0원

0%

업데이트

매매진행상황

진행상황	종목코드	종목명	매수조건식	주문량	매수량	매수가	매수금	매도가	현재가	손익률
매수완료	046120	오르비텍	condition	780	780	6404	4995120		6390	-1.15
매수완료	278650	노터스	condition	600	600	8329	4997400		8350	-0.68
매수중	380320	삼성머스트스...	condition	868	29	5760	167040		5770	-0.75
매수완료	005930	삼성전자	condition	84	84	59100	4964400		59100	-0.93
매수완료	246690	TS인베스트먼트	condition	1666	1666	3000	4998000		3005	-0.76
매수완료	018000	유니슨	condition	1848	1848	2705	4998840		2705	-0.93
매수중	362320	청담글로벌	condition	467	417	10700	4461900		10700	-0.93
매수중	024810	이화전기	condition	3984	900	1250	1125000		1250	

매매전략

계좌번호	매수조건식	매수가	종투자금	매수종목수	평균당투자금	익절	익절률	손절	손절률	위소
8024053711	condition	00	50000000	10	5000000	True	3.3	True	-4	취소

자동매매

잔고매도

관심종목

매수전략

매수조건식

condition2

☐ 관심종목만 매수

☐ 시장가
 ☒ 지정가

00:현재가

종투자금

50000000

매수종목 수

10

종목 당 투자금 :

5000000 원

매도전략

☒ 익절

매수단가 대비

3.30

☒ 시장가
 ☐ 지정가

☒ 손절

매수단가 대비

-4.00

전략생성

주문

주문	제결	미제결	잔고		
주문번호	계좌번호	시간	종목코드	종목명	주문량
0107648	8024053711	104314	A046120	오르비텍	780
0107664	8024053711	104316	A278650	노터스	600
0107677	8024053711	104317	A380320	삼성머스트스...	868
0107740	8024053711	104326	A005930	삼성전자	84
0107768	8024053711	104328	A246690	TS인베스트먼트	1666

주문내역창

보유종목

종목코드	종목명	보유수량	평균단가	손익률	평가금액
A000270	기아	41	80,849.00	-0.38	3,333,200
A005380	현대차	18	183,500.00	-0.39	3,321,000
A005930	삼성전자	49	59,269.00	-1.21	2,895,900
A219550	디와이디	28	2,398.00	2.07	69,160

기존보유종목

진행상황

조건식에 의한
자동매매현황

주문내역창

기존보유종목

- 3. 기능소개 -컨트롤창(자동매매창)

사용자 조건식

자동매매 잔고매도 관심종목

매수전략

매수조건식: condition2

☐ 관심종목만 매수

☐ 시장가 ☒ 지정가 00:현재가

총투자금: 50000000

매수종목 수: 10

종목 당 투자금: 5000000 원

매도전략

☒ 익절 매수단가 대비: 3.30

☒ 시장가 ☐ 지정가

☒ 손절 매수단가 대비: -4.00

전략생성

00:현재가
05:조건부지정가
06:최유리지정가
07:최우선지정가
10:지정가IOC
13:시장가IOC
16:최유리IOC
20:지정가FOK
23:시장가FOK
26:최유리FOK
61:장전시간외종가
62:시간외단일가매매
81:장후시간외종가

-키움증권 HTS내에서 설정한 조건검색식을 통한 종목 정보를 매수, 매도 전략을 설정하는 창

- 매수평단가 대비 익절/손절 %를 지정할 수 있으며 매매 방식을 시장가 또는 지정가로 설정 가능

-전략생성 버튼 클릭 시 매매현황 창이 업데이트 되고 조건식에 새로 편입되는 종목을 감시하여 매수주문을 요청합니다.

- 3. 기능소개 -컨트롤창(잔고매도)

자동매매

잔고매도

관심종목

매도종목정보

종목코드

005380

종목명

현대차

수량

18

평균단가

183500

매도전략

☒ 익절

매수단가 대비

3.30

☒ 시장가

☐ 지정가

☒ 손절

매수단가 대비

-4.00

매도등록

- 기존 보유종목에 대한 컨트롤창
- 보유종목창에서 종목 선택 시 종목코드 및 종목명을 출력하며 원하는 수량을 매도 등록할수 있다.
- 매도 등록 버튼 클릭시 매매진행상황창에 등록되며 설정한 손/익절 포인트에 도달하면 매도주문을 포함

보유종목						
종목코드	종목명	보유수량	평균단가	손익률	평가금액	
A000270	기아	41	80,849.00	-0.50	3,329,200	3,
A005380	현대차	18	183,500.00	-0.66	3,312,000	3,
A219550	디와이디	28	2,398.00	1.44	68,740	67,
A366030	공구우먼	131	75,928.00	-4.63	9,576,100	9,

- 3. 기능소개 - 주문내역(주문, 잔고)

주문

주문	체결	미체결	잔고		
주문번호	계좌번호	시간	종목코드	종목명	주문량
0107648	8024053711	104314	A046120	오르비텍	780
0107664	8024053711	104316	A278650	노터스	600
0107677	8024053711	104317	A380320	삼성머스트스...	868
0107740	8024053711	104326	A005930	삼성전자	84
0107768	8024053711	104328	A246690	TS인베스트먼트	1666

주문

- 키움API를 통해 연결된 계좌의 주문현황을 실시간으로 받아서 표시

주문

주문		체결	미체결		잔고
계좌번호	종목코드	종목명	보유수량	주문가능수량	매입단가
8024053711	A000270	기아	41	41	80849
8024053711	A005380	현대차	18	18	183500
8024053711	A005930	삼성전자	84	84	59162
8024053711	A219550	디와이디	2077	2077	2439
8024053711	A366030	공구우먼	131	131	75928

잔고

- 프로그램 실행시 기존 잔고와, 자동매매에 의한 잔고정보를 실시간 업데이트

- 3. 기능소개 - 주문내역(체결, 미체결)

주문

주문	체결	미체결	잔고
----	----	-----	----

주문번호	체결시간	종목코드	종목명	주문량	단위체결량
0107648	104314	A046120	오르비텍	780	1
0107648	104314	A046120	오르비텍	780	1
0107648	104314	A046120	오르비텍	780	1
0107648	104315	A046120	오르비텍	780	1
0107648	104315	A046120	오르비텍	780	1

체결

- 주문에 대한 체결현황을 실시간으로 받아온다.

미체결

- 주문에 대한 미체결 현황을 실시간으로 받아온다.

주문

주문		체결	미체결		잔고
주문번호	종목코드	종목명	주문수량	미체결량	매매구분
0132259	A219550	디와이디	2049	2049	-매도
0132263	A099220	SDN	1736	1497	-매도
0132270	A035460	기산텔레콤	1109	1109	-매도

- 3. 기능소개 - 보유종목

종목코드	종목명	보유수량	평균단가	손익률	평가금액	
A000270	기아	41	80,849.00	-0.74	3,321,000	3,
A005380	현대차	18	183,500.00	-0.39	3,321,000	3,
A219550	디와이디	28	2,398.00	0.21	67,900	67,
A366030	공구우먼	131	75,928.00	-5.16	9,523,700	9,

평균단가	손익률	평가금액	매입금액	손익금액	청산
80,849.00	-0.87	3,316,900	3,314,809	-28,738	청산
183,500.00	-0.39	3,321,000	3,303,000	-12,818	청산
2,398.00	0.21	67,900	67,144	140	청산
75,928.00	-5.29	9,510,600	9,946,568	-525,932	청산

- 자동매매에 해당되지 않는 종목들에 대한 컨트롤창
- 잔고매도를 통해 매도전략을 설정할수 있음
- 잔고매도가 아닌 청산 클릭시 현재가 기준 시장가로 청산

- 3. 기능소개 - 매매진행상황

진행상황	종목코드	종목명	매수조건식	주문량	매수량	매수가	매수금	매도가	현재가	손익률
매도완료	046120	오르비텍	condition	780	780	6404	4995120	6200	6220	-4.1
매도완료	362320	청담글로볼	condition	467	467	10700	4996900	10355	10500	-4.18
매도완료	380320	삼성머스트스...	condition	868	868	5760	4999680	5992	5870	2.69
매도완료	024810	이화전기	condition	3984	3984	1250	4980000	1211	1180	-3.32
매도완료	340350	SK6호스택	condition	2173	2173	2300	4997900	2221	2280	-4.17
매도완료	018000	유니슨	condition	1848	1848	2705	4998840	2625	2605	-4.05
매도완료	332290	누보	condition	1620	1620	3085	4997700	2977	2975	-4.31
매도완료	215100	로보로보	condition	632	632	7880	4980160	7661	7260	-4.46
매수완료	278650	노터스	condition	600	600	8329	4997400		8250	-1.87
매수완료	214610	미코바이오메드	condition	331	331	15093	4995783		14850	-2.53
매수완료	005930	삼성전자	condition	84	84	59100	4964400		58900	-1.27
매수완료	189980	한국에프엔비	condition	1644	1644	3031	4982964		3005	-1.78
매수완료	091990	셀트리온헬스...	condition	73	73	67526	4929398		67500	-0.97
매도완료	246690	TS인베스트먼트	condition	1666	1666	3000	4998000	2950	2955	-2.59
매도완료	035460	기산탈러콤	condition	1109	1109	4502	4992718	4486	4505	-1.31
매도완료	099220	SDN	condition	1736	1736	2880	4999680	2899	2920	-0.24
매도완료	219550	디와이디	condition	2049	2049	2440	4999560	2435	2430	-1.13
매도완료	036120	SCI평가정보	condition	1721	1721	2900	4990900	2869	3050	-1.96
매수완료	049480	오픈베이스	condition	1739	1739	2875	4999625		2875	-0.93
매도완료	063440	SM Life Design	condition	2164	2164	2305	4988020	2245	2190	-3.3
매수완료	006910	보성파워텍	condition	880	880	5673	4992240		5650	-1.33

주문량	매수량	매수가	매수금	매도가	현재가	손익률	평가손익	매수시간	매도시간	청산
2	632	7880	4980160	7661	7260	-4.46	-222247	2022-06-28 10:5...	2022-06-28 10:5...	매도완료
0	600	8329	4997400		8250	-1.87	-93595	2022-06-28 10:4...		청산
1	331	15093	4995783		14850	-2.53	-126418	2022-06-28 10:4...		청산
	84	59100	4964400		58900	-1.27	-62859	2022-06-28 10:4...		청산
44	1644	3031	4982964		3000	-1.95	-97008	2022-06-28 10:5...		청산
	73	67526	4929398		67500	-0.97	-47721	2022-06-28 10:5...		청산
66	1666	3000	4998000	2950	2960	-2.59	-129294	2022-06-28 10:4...	2022-06-28 11:3...	매도완료
09	1109	4502	4992718	4486	4505	-1.31	-65163	2022-06-28 10:4...	2022-06-28 11:3...	매도완료
36	1736	2880	4999680	2899	2955	-0.24	-11969	2022-06-28 11:1...	2022-06-28 11:3...	매도완료
49	2049	2440	4999560	2435	2440	-1.13	-56670	2022-06-28 11:2...	2022-06-28 11:3...	매도완료
21	1721	2900	4990900	2869	3070	-1.96	-97730	2022-06-28 11:3...	2022-06-28 11:3...	매도완료
39	1739	2875	4999625		2870	-1.1	-55124	2022-06-28 11:3...		청산
64	2164	2305	4988020	2245	2135	-3.3	-164709	2022-06-28 11:3...	2022-06-28 11:4...	매도완료
0	880	5673	4992240		5650	-1.33	-66546	2022-06-28 11:3...		청산
66	1466	3410	4999060		3325	-3.41	-170371	2022-06-28 11:3...		청산
4	244	20444	4988336	19776	19850	-4.06	-202598	2022-06-28 11:3...	2022-06-28 11:4...	매도완료
6	286	17407	4978402	17550	17500	-0.11	-5626	2022-06-28 11:4...	2022-06-28 11:4...	매도완료
7	287	17361	4982607	17300	17300	-1.28	-63727	2022-06-28 11:4...	2022-06-28 11:4...	매도완료
8	338	14750	4985500	14551	14600	-1.94	-96760	2022-06-28 11:4...	2022-06-28 11:4...	매도완료
61	1461	3416	4990776	3429	3425	-0.67	-33330	2022-06-28 11:4...	2022-06-28 11:4...	매도완료
	0	0	0					2022-06-28 11:4...		주문취소

- 자동매매 진행현황 창
- 종목당 매수가, 매도가, 손익률 등을 확인할 수 있음
- 청산 클릭 시 현재가 기준 시장가로 청산
- 매수주문이 완료되지 않은 주문은 주문 취소 가능

- 3. 기능소개 - 계좌 정보

예수금	D+2추정예수금(예탁자산평가액)	당일투자원금	당일투자손익	당일손익률
210,662,273원	211,523,784원	0원	0원	0%
				업데이트

예수금	D+2추정예수금(예탁자산평가액)	당일투자원금	당일투자손익	당일손익률
210,662,273원	162,140,965원	96,366,615원	-2,280,588원	-2.33%
				업데이트

- 간단한 계좌 정보를 볼 수 있다.
- 업데이트 클릭 시 실시간 계좌 정보를 가져옴

- 4. SourceCode – 사용된 기본 List 정보

```
List<Condition> conditionList = new List<Condition>(); //조건식리스트

List<BalanceSellStrategy> balanceSellStrategyList = new List<BalanceSellStrategy>(); // 보유 종목 매도 전략 리스트
List<TradingStrategy> tradingStrategyList = new List<TradingStrategy>(); //매매전략 리스트
List<SettlementItem> settleItemList = new List<SettlementItem>(); //진행중인 청산 종목 리스트

List<TradingItem> tryingOrderList = new List<TradingItem>(); //주문 접수 시도 종목 리스트
List<SettlementItem> tryingSettleList = new List<SettlementItem>(); //청산 접수 시도 종목 리스트
List<BalanceSellStrategy> tryingSellList = new List<BalanceSellStrategy>(); //잔고 매도 접수 시도 종목 리스트

List<TradingStrategy> addAccountBalanceList = new List<TradingStrategy>(); //매매전략 취소 시 주문중인 아이템 리스트

List<StockItem> stockItemList = new List<StockItem>(); //상장종목리스트

bool marketClose = false; //true가 되면 모든 종목 청산

//List<CheckItem> checkitemList = new List<CheckItem>();
//DBCon dbcon = new DBCon();

double TAX = 0.0023;

double FEE = 0.00015; //모투의경우 0.0035

int screenNum = 1000;
string currentAccount = String.Empty;
long DEPOSIT = 0;
```

- 4. SourceCode – 이벤트 목록

```
InitializeComponent();

loginToolStripMenuItem.Click += ToolStripMenuItem_Click; //로그인 메뉴 Event

createStrategyButton.Click += Button_Click;
balanceSellButton.Click += Button_Click;
AddInterestButton.Click += Button_Click;
SetaccountbalanceBtn.Click += Button_Click;

accountComboBox.SelectedIndexChanged += ComboBox_SelectedIndexChanged;

accountBalanceDataGridView.CellClick += DataGridView_CellClick;
autoTradingDataGridView.CellClick += DataGridView_CellClick;
tradingStrategyDataGridView.CellClick += DataGridView_CellClick;

accountBalanceDataGridView.SelectionChanged += DataGridView_SelectionChanged;

interestConditionListBox.SelectedIndexChanged += ListBox_SelectedIndexChanged;

axKHOpenAPI1.OnEventConnect += API_OnEventConnect; //로그인 결과 Event
axKHOpenAPI1.OnReceiveConditionVer += API_OnReceiveConditionVer; // 조건식 리스트 요청 결과 Event
axKHOpenAPI1.OnReceiveRealCondition += API_OnReceiveRealCondition; //실시간 조건식 편입/이탈 정보 Event
axKHOpenAPI1.OnReceiveTrData += API_OnReceiveTrData; //Tr요청 결과 Event
axKHOpenAPI1.OnReceiveChejanData += API_OnReceiveChejanData; // 실시간 체결/잔고 정보 Event
axKHOpenAPI1.OnReceiveRealData += API_OnReceiveRealData; //실시간 데이터
axKHOpenAPI1.OnReceiveTrCondition += API_OnReceiveTrCondition; //전략생성시 데이터
axKHOpenAPI1.OnReceiveMsg += API_OnReceiveMsg;

tabControl2.DrawMode = TabDrawMode.OwnerDrawFixed;
tabControl2.DrawItem += new DrawItemEventHandler(this.tabControl2_DrawItem);

tabControl1.DrawMode = TabDrawMode.OwnerDrawFixed;
tabControl1.DrawItem += new DrawItemEventHandler(this.tabControl1_DrawItem);

//form 종료 이벤트
this.FormClosing += Form_Closing;

// tabControl Color
autoTradingDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
tradingStrategyDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
accountBalanceDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
orderDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
conclusionDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
outstandingDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;
balanceDataGridView.ColumnHeadersDefaultCellStyle.BackColor = col;

//tabControl size
tabControl2.SizeMode = TabSizeMode.Fixed;
tabControl1.SizeMode = TabSizeMode.Fixed;
```

- 4. SourceCode – 전략생성

```
TradingStrategy ts = new TradingStrategy(account, buyingCondition, buyingOrderOption, totalInvestment, itemCount, usingTakeProfit, takeProfitRate, sellOrderOption, usingStopLoss, stopLossRate, onlyIntest);  
/*ts.usingCondition = true;*/  
tradingStrategyList.Add(ts);
```

```
참고 27%  
public class TradingStrategy  
{  
    public string account; //매매 계좌  
    public Condition buyingCondition; //매수 조건식  
    public string buyingOrderOption; //매수 현재가 or 시장가 등등 호가 옵션  
    public long totalInvestment; //총 투자금  
    public int buyingItemCount; //매수 종목수  
    public int remainingItemCount; // 남은 매수할 종목 수  
  
    public long itemInvestment; //종목별 투자금  
    public bool onlyIntrest;  
  
    public bool usingTakeProfit; //익절 사용여부  
    public double takeProfitRate; //익절률  
    public string sellOrderOption; //매도 현재가 or 시장가 등등 호가 옵션  
    public bool usingStopLoss; //손절 사용 여부  
    public double stopLossRate; //손절률  
  
    public bool usingCondition; //조건식 사용여부  
  
    //매매 진행 종목 리스트  
    public List<TradingItem> tradingItemList = new List<TradingItem>();  
}
```

- 전략 생성 시 tradingStrategyList에 전략이 추가 되고 자동매매를 실시함

- 4. SourceCode – 편입종목매수

```
//Console.WriteLine("남은 매수 가능 종목 수 : " + ts.remainingItemCount);
if (ts.remainingItemCount > 0) // 매수 가능 종목 수 > 0
{
    StockItem stockItem = stockItemList.Find(o => o.Code.Equals(itemCode));

    if (!ts.onlyIntrest || (ts.onlyIntrest && ts.buyingCondition.interestItemList.Contains(stockItem)))
    {
        TradingItem ti = ts.tradingItemList.Find(o => o.itemCode.Contains(itemCode));

        //매매 조건식 매매종목리스트에 편입 종목이 존재하지 않을 경우
        if (ti == null)
        {
            //Console.WriteLine("종목매수 : " + axKHOpenAPI1.GetMasterCodeName(itemCode));
            Console.WriteLine("매수가능종목수 : " + ts.remainingItemCount.ToString());
            ts.remainingItemCount--; //남은 매수가능 종목수 -1

            //여기서 매수할 종목 가격 요청
            axKHOpenAPI1.SetInputValue("종목코드", itemCode);
            axKHOpenAPI1.CommRqData("매수종목정보요청;" + ts.buyingCondition.index, "opt10001", 0, GetScreenNum());
        }
    }
}
```

조건식에 새로운 종목이 편입되면 해당종목 매수여부 판단후 가격정보를 요청

```
private void API_OnReceiveTrData(object sender, AxKHOpenAPILib._DKHOpenAPIEvents_OnReceiveTrDataEvent e)
{
    Console.WriteLine("sRQName,TR이름 : " + e.sRQName + e.sTrCode);

    if (e.sRQName.Contains("매수종목정보요청"))
    {
        string[] rqName = e.sRQName.Split(';');
        int conditionIndex = int.Parse(rqName[1]);

        TradingStrategy ts = tradingStrategyList.Find(o => o.buyingCondition.index == conditionIndex)
        if (ts != null /*&& !existItemList.Contains(itemCode)*/)
        {
            string itemCode = axKHOpenAPI1.GetCommData(e.sTrCode, e.sRQName, 0, "종목코드").Trim();
            string price = axKHOpenAPI1.GetCommData(e.sTrCode, e.sRQName, 0, "현재가").Trim();
            //Console.WriteLine("매수종목정보요청 가격 - " + price);

            try
            {
                int cPrice = Math.Abs(int.Parse(price));
                int qnt = (int)(ts.itemInvestment / cPrice);
                if (cPrice > 0)
                {
                    int orderResult = axKHOpenAPI1.SendOrder(
                        "편입종목매수;" + ts.buyingCondition.name + ";" + itemCode,
                        GetScreenNum(),
                        ts.account, //계좌번호
                        1, //신규매수
                        itemCode,
                        qnt,
                        cPrice,
                        ts.buyingOrderOption,
                        ""
                    );
                }
            }
            catch { }
        }
    }
}
```

- 매수종목 가격정보를 가지고 매수주문요청

- 4. - 편입종목매수

```
//Console.WriteLine("남은 매수 가능 종목 수 : " + ts.remainingItemCount);
if (ts.remainingItemCount > 0) // 매수 가능 종목 수 > 0
{
    StockItem stockItem = stockItemList.Find(o => o.Code.Equals(itemCode));

    if (!ts.onlyIntrest || (ts.onlyIntrest && ts.buyingCondition.interestItemList.Contains(stockItem)))
    {
        TradingItem ti = ts.tradingItemList.Find(o => o.itemCode.Contains(itemCode));

        //매매 조건식 매매종목리스트에 편입 종목이 존재하지 않을 경우
        if (ti == null)
        {
            //Console.WriteLine("종목매수 : " + axKHOpenAPI1.GetMasterCodeName(itemCode));
            Console.WriteLine("매수가능종목수 : " + ts.remainingItemCount.ToString());
            ts.remainingItemCount--; //남은 매수가능 종목수 -1

            //여기서 매수할 종목 가격 요청
            axKHOpenAPI1.SetInputValue("종목코드", itemCode);
            axKHOpenAPI1.CommRqData("매수종목정보요청;" + ts.buyingCondition.index, "opt10001", 0, GetScreenNum());
        }
    }
}
```

조건식에 새로운 종목이 편입되면 해당종목 매수여부 판단후 가격정보를 요청

```
private void API_OnReceiveTrData(object sender, AxKHOpenAPILib._DKHOpenAPIEvents_OnReceiveTrDataEvent
{
    Console.WriteLine("sRQName,TR이름 : " + e.sRQName + e.sTrCode);

    if (e.sRQName.Contains("매수종목정보요청"))
    {
        string[] rqName = e.sRQName.Split(';');
        int conditionIndex = int.Parse(rqName[1]);

        TradingStrategy ts = tradingStrategyList.Find(o => o.buyingCondition.index == conditionIndex)
        if (ts != null /*&& !existItemList.Contains(itemCode)*/)
        {
            string itemCode = axKHOpenAPI1.GetCommData(e.sTrCode, e.sRQName, 0, "종목코드").Trim();
            string price = axKHOpenAPI1.GetCommData(e.sTrCode, e.sRQName, 0, "현재가").Trim();
            //Console.WriteLine("매수종목정보요청 가격 - " + price);

            try
            {
                int cPrice = Math.Abs(int.Parse(price));
                int qnt = (int)(ts.itemInvestment / cPrice);
                if (cPrice > 0)
                {
                    int orderResult = axKHOpenAPI1.SendOrder(
                        "편입종목매수;" + ts.buyingCondition.name + ";" + itemCode,
                        GetScreenNum(),
                        ts.account, //계좌번호
                        1, //신규매수
                        itemCode,
                        qnt,
                        cPrice,
                        ts.buyingOrderOption,
                        ""
                    );
                }
            }
            catch { }
        }
    }
}
```

- 매수종목 가격정보를 가지고 매수주문요청

- 4. SourceCode - 보유종목 잔고매도

```
private void API_OnReceiveRealData(object sender, AxKHOpenAPILib._DKHOpenAPIEvt
{
    string itemCode = e.sRealKey.Trim();
    //Console.WriteLine("실시간데이터 : " + itemCode);
    if (e.sRealType.Equals("주식세력"))
    {
        string cPrice = axKHOpenAPI1.GetCommRealData(itemCode, 10);
        string lowPrice = axKHOpenAPI1.GetCommRealData(itemCode, 18);
        string openPrice = axKHOpenAPI1.GetCommRealData(itemCode, 16);
        string mesurate = axKHOpenAPI1.GetCommRealData(itemCode, 129);

        long cPrice_long = Math.Abs(long.Parse(cPrice)); //현재가

        //종목 -> 매매 전략 검색
        foreach (TradingStrategy ts in tradingStrategyList)
        {
            //실시간 종목에 대한 매매 전략

            BalanceSellStrategy bss = balanceSellStrategyList.Find(o => o.itemCode == itemCode);
            if (bss != null)
            {
                if (!bss.isSold && bss.buyingPrice != 0)
                {
                    //double profitRate = 100 * (cPrice_long - bss.buyingPrice) / bss.buyingPrice;
                    double profitRate = (((cPrice_long * bss.sellQnt) * (1.00 - TA
                    Console.WriteLine("손익률 : " + Math.Round(profitRate, 2));

                    if (bss.takeProfitRate <= Math.Round(profitRate, 2)) //익절 만족
                    {
                        //매도주문
                        int orderResult = axKHOpenAPI1.SendOrder(
                            "잔고익절매도;" + bss.itemCode,
                            GetScreenNum(),
                            bss.account, //계좌번호
                            2, //신규매수
                            itemCode,
                            (int)bss.sellQnt,
                            (int)cPrice_long,
                            "00",
                            ""
                        );
                    }
                }
                else if (bss.stopLossRate >= Math.Round(profitRate, 2)) //손절 만족
                {
                    //매도주문
                    int orderResult = axKHOpenAPI1.SendOrder(
                        "잔고손절매도;" + bss.itemCode,
                        GetScreenNum(),
                        bss.account, //계좌번호
                        2, //신규매수
                        itemCode,
                        (int)bss.sellQnt,
                        0,
                        "03",
                        ""
                    );
                }
            }
        }
    }
}
```

- OnReceiveRealData 이벤트로 실시간 세세를 받아
옴

- 손절/익절 포인트 도달 시 주문요청

- 4. SourceCode – 체결/미체결 정보

```
private void API_OnReceiveChejanData(object sender, AxKHOpenAPILib._DKHOpenAPIEvents_OnReceiveChejanDataEvent e)
{
    //전달 정보: 특정 주문에 대한 접수/체결/잔고
    //전달되지 않는 정보: 어떤 매매전략으로 한 주문인가?

    if (e.sGubun.Equals("0")) //접수 혹은 체결

    {
        //Console.WriteLine("접수 or 체결");
        string account = axKHOpenAPI1.GetChejanData(9201); //계좌번호
        string orderNumber = axKHOpenAPI1.GetChejanData(9203); //주문번호
        string itemCode = axKHOpenAPI1.GetChejanData(9001); //종목코드
        string orderState = axKHOpenAPI1.GetChejanData(913); //주문상태
        string itemName = axKHOpenAPI1.GetChejanData(302).Trim(); //종목명
        string orderQuantity = axKHOpenAPI1.GetChejanData(900); //주문수량
        string orderPrice = axKHOpenAPI1.GetChejanData(904); //주문가격
        string outstanding = axKHOpenAPI1.GetChejanData(902); //미체결수량
        string orderType = axKHOpenAPI1.GetChejanData(905); //주문구분
        string tradingType = axKHOpenAPI1.GetChejanData(906); //매매구분
        string time = axKHOpenAPI1.GetChejanData(908); //주문 혹은 체결시간
        string conclusionPrice = axKHOpenAPI1.GetChejanData(910); //누적평균체결가
        string conclusionPrice_unit = axKHOpenAPI1.GetChejanData(914); //단위 체결가
        string conclusionQuantity = axKHOpenAPI1.GetChejanData(911); //체결량
        string unitConclusionQty = axKHOpenAPI1.GetChejanData(915); //단위체결량
        string price = axKHOpenAPI1.GetChejanData(10); //현재가
        string orderNumber_pre = axKHOpenAPI1.GetChejanData(904); //원주문번호 주문위스, 정정시 사용

        Console.WriteLine("주문량 : " + orderQuantity + " 체결가 : " + conclusionPrice + " 체결량 : " + conclusionQuantity + " 미체결량 : " + outstanding);
        Console.WriteLine("주문상태 : " + orderState + " 주문구분 : " + orderType + " 매매구분 : " + tradingType);
        Console.WriteLine("주문번호 : " + orderNumber + " 원주문번호 : " + orderNumber_pre);

        if (orderState.Equals("접수"))
        {
            //주문위스가 자동매매종 손/익절을 위한 주문위스인지 사용자컨트롤에 의한 주문위스인지... check
            //잔자의 경우 손/익절 이후 remainingItemCount++, 후자의 경우 주문위스로 remainingItemCount++
            if (orderType.Equals("+매수") && int.Parse(outstanding) == 0 && conclusionQuantity.Equals("")) //체결량이 존재할때 주문위스 미체결클리어신호
            {
                //매매전략 업데이트
                foreach (TradingStrategy ts in tradingStrategyList)
                {
                    //미체결 dataGridView 정보 update

                    foreach (DataGridViewRow row in outstandingDataGridView.Rows)
                    {
                        foreach (TradingStrategy ts in tradingStrategyList)
                        {
                            TradingItem ti = ts.tradingItemList.Find(s => s.buyOrderNum.Equals(orderNumber));

                            if (ti != null && !ti.medoCondition.Equals("사용자"))
                            {
                                //잔량에 대한 익절/손절주문
                                //매도주문
                                if (DialogResult == 0)
                                {
                                    Console.WriteLine("testtttt");
                                    Console.WriteLine("계좌번호 : " + account);
                                    Console.WriteLine("매도수량 : " + ti.buyingQty);
                                    Console.WriteLine("종목코드 : " + ti.itemCode);
                                    Console.WriteLine("가격 : " + Math.Abs(int.Parse(price)));

                                    int orderResult = axKHOpenAPI1.SendOrder(
                                        "종목" + ti.medoCondition + "매도",
                                        GetScreenNum(),
                                        account, //계좌번호
                                        2, //신규매도
                                        ti.itemCode,
                                        ti.buyingQty,
                                        Math.Abs(int.Parse(price)),
                                        ts.sellOrderOption,
                                        ""
                                    );
                                    Console.WriteLine("손/익절 주문결과 : " + orderResult);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

- OnReceiveChejanData 이벤트로 실시간 체결정보를 받아옴
- 주문중에 손/익절 포인트 도달 시 주문 취소후 나머지 손/익절 실행

- 4. SourceCode – 장마감청산

```
private void TimerFunction(object sender, EventArgs e)
{
    string now = DateTime.Now.ToString("HH:mm:ss");

    //3시되면 1.전략삭제, 2.미체결 취소, 3.잔고 매도
    if (now.Contains("15:00:00")) //3시되면
    {
        Console.WriteLine("start!!!");
        //1. 전략 삭제
        tradingStrategyList.Clear();
        tradingStrategyDataGridView.Rows.Clear();

        //2. 모든 미체결 주문 취소
        foreach (DataGridViewRow rows in outstandingDataGridView.Rows)
        {
            int orderResult = axKHOpenAPI1.SendOrder("장마감청산",
                GetScreenNum(),
                currentAccount,
                3,
                rows.Cells["미체결_종목코드"].Value.ToString().Replace("A", "").Trim(),
                0,
                0,
                "",
                rows.Cells["미체결_주문번호"].Value.ToString());

            Console.WriteLine("장마감주문취소결과 : " + orderResult);
            Console.WriteLine("장마감청산 종목코드 : " + rows.Cells["미체결_종목코드"].Value.ToString());
        }

        marketClose = true;
    }

    if(marketClose == true)
    {
        //잔고 모두 청산
        foreach(DataGridViewRow row in balanceDataGridView.Rows)
        {
            int orderResult = axKHOpenAPI1.SendOrder("청산매도주문",
                GetScreenNum(),
                currentAccount,
                2,
                row.Cells["잔고_종목코드"].Value.ToString().Replace("A", "").Trim(),
                int.Parse(row.Cells["잔고_주문가능수량"].Value.ToString()),
                0,
                "03",
                "");
        }
    }
}
```

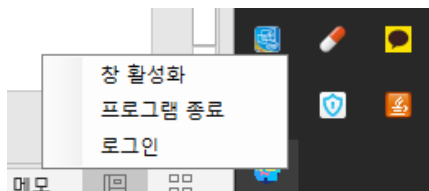
- Winform timer를 이용해 15:00가 되면 자동매매 중단 후 모든 잔고 청산

- 4. SourceCode – TrayIcon

```
private void Form_Closing(object sender, FormClosingEventArgs e)
{
    e.Cancel = true;
    this.Visible = false;
}

private void trayItem_clicked(object sender, ToolStripItemClickedEventArgs e)
{
    if(e.ClickedItem.Tag != null)
    {
        if (e.ClickedItem.Tag.ToString().Equals("EXIT"))
        {
            this.Close();
            this.Dispose();
            Properties.Settings.Default.Save();
            Application.Exit();
        }
        else if (e.ClickedItem.Tag.ToString().Equals("OPEN"))
        {
            this.Visible = true;
        }
        else if (e.ClickedItem.Tag.ToString().Equals("ACCOUNT"))
        {
            if (e.ClickedItem.Text.Equals("로그인"))
            {
                axKHOpenAPI1.CommConnect();
            }
        }
    }
}
```

- 자동매매 진행 중 실수로 프로그램 종료 클릭시 종료 방지



- 5. 끝

- <https://github.com/JongWon112/AutoTradingSystem/blob/main/EolabAutoTradingSystem/Form1.cs>