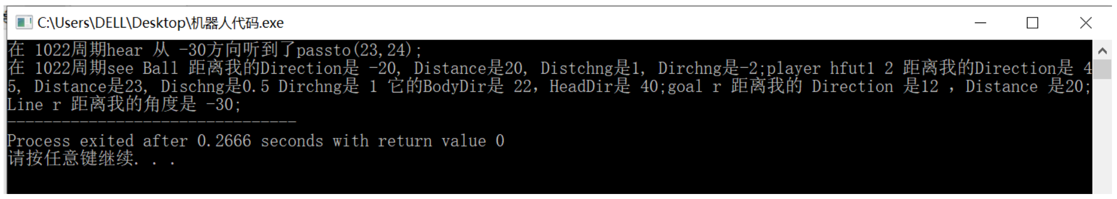
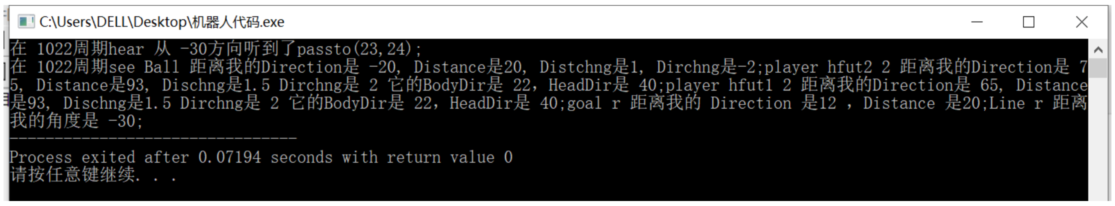
作者：计科 19-4 曾宇杰
学号：2019218164

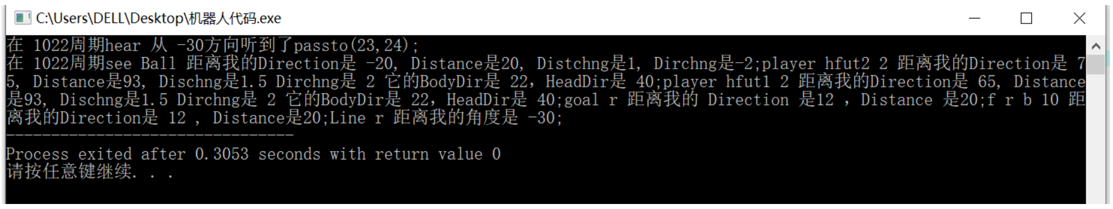
运行结果截图：
输入：(hear 1022 -30 passto(23,24))(see 1022 ((ball) -20 20 1 -2) ((player hfut1 2) 45 23 0.5 1 22 40) ((goal r) 12 20) ((Line r) -30))


```
C:\Users\DELL\Desktop\机器人代码.exe
在 1022周期hear 从 -30方向听到了passto(23,24);
在 1022周期see Ball 距离我的Direction是 -20, Distance是20, Distchng是1, Dirchng是-2;player hfut1 2 距离我的Direction是 4
5, Distance是23, Dischng是0.5 Dirchng是 1 它的BodyDir是 22, HeadDir是 40;goal r 距离我的 Direction 是12 , Distance 是20;
Line r 距离我的角度是 -30;
--------------------------------
Process exited after 0.2666 seconds with return value 0
请按任意键继续. . .
```

输入：(hear 1022 -30 passto(23,24))(see 1022 ((ball) -20 20 1 -2) ((player hfut2 2) 75 93 1.5 2 22 40) ((player hfut1 2) 65 93 1.5 2 22 40) ((goal r) 12 20) ((Line r) -30))


```
C:\Users\DELL\Desktop\机器人代码.exe
在 1022周期hear 从 -30方向听到了passto(23,24);
在 1022周期see Ball 距离我的Direction是 -20, Distance是20, Distchng是1, Dirchng是-2;player hfut2 2 距离我的Direction是 7
5, Distance是93, Dischng是1.5 Dirchng是 2 它的BodyDir是 22, HeadDir是 40;player hfut1 2 距离我的Direction是 65, Distance
是93, Dischng是1.5 Dirchng是 2 它的BodyDir是 22, HeadDir是 40;goal r 距离我的 Direction 是12 , Distance 是20;Line r 距离
我的角度是 -30;
--------------------------------
Process exited after 0.07194 seconds with return value 0
请按任意键继续. . .
```

输入：(hear 1022 -30 passto(23,24))(see 1022 ((ball) -20 20 1 -2) ((player hfut2 2) 75 93 1.5 2 22 40) ((player hfut1 2) 65 93 1.5 2 22 40) ((goal r) 12 20) ((f r b 10) 12 20) ((Line r) -30))


```
C:\Users\DELL\Desktop\机器人代码.exe
在 1022周期hear 从 -30方向听到了passto(23,24);
在 1022周期see Ball 距离我的Direction是 -20, Distance是20, Distchng是1, Dirchng是-2;player hfut2 2 距离我的Direction是 7
5, Distance是93, Dischng是1.5 Dirchng是 2 它的BodyDir是 22, HeadDir是 40;player hfut1 2 距离我的Direction是 65, Distance
是93, Dischng是1.5 Dirchng是 2 它的BodyDir是 22, HeadDir是 40;goal r 距离我的 Direction 是12 , Distance 是20;f r b 10 距
离我的Direction是 12 , Distance是20;Line r 距离我的角度是 -30;
--------------------------------
Process exited after 0.3053 seconds with return value 0
请按任意键继续. . .
```

源代码：
```cpp
#ifndef FIRST_HOMEWORK_OF_ROBOCUP
#define FIRST_HOMEWORK_OF_ROBOCUP
#include <iostream>
#include <string>
#include <vector>

#endif // !FIRST_HOMEWORK_OF_ROBOCUP
using namespace std;
class message_manager
{
private:

    string messages;                    //消息原始字符串
    string messages_see;
    string messages_hear;
    vector<string> HEAR;               //经过处理后的消息分段存入 vector 中
    vector<string> SEE;

    string time;                        //周期

    string sender;                      //听到的信息
    string message;

    vector<string> ball;                //看到的信息
    vector<string> flag;
    vector<string> Line;
    vector<string> goal;
    vector<string> teammates;
    vector<string> opponent;


public:
    message_manager(string message);
    ~message_manager();
    void message_print();
};

/**
 * description : 将原始消息切割分类存入小数组并且实现格式化输出
 */
void message_manager::message_print()
{
    cout << "在 " << time << "周期
" << *(HEAR.begin()) << " 从 " << sender << "方向听到了
" << message << ";" << endl;
    cout << "在 " << time << "周期" << *(SEE.begin());
    vector<string>::iterator itt = SEE.begin()+2;

    while (itt != SEE.end())
    {
        vector<string> container;
        int first = (*itt).find('(')+2;
        int end = (*itt).find(')');
        string message_name = (*itt).substr(first, end - first);

        container.push_back(message_name);

        int index = end+1;
        while (index < (*itt).size())                             //现
将原始消息切割成小段方便但会输出
```

```
        {
            string temp = "";
            while ((*itt)[index] != ' ' && (*itt)[index] != '(' && (*it
t)[index] != ')' && index < (*itt).size())
            {
                temp.append((*itt).substr(index, 1));
                index++;
            }
            if (temp != "")
            {
                container.push_back(temp);
            }

            index++;
        }

        if ((*itt).find("ball") != (*itt).npos)                 //将 SEE
到的信息存储到对应数组并且格式化输出
        {
            ball.push_back(*itt);                               //储存原
始消息字符串到相应数组

            cout << " Ball 距离我的 Direction
是 " << container[1] << ", Distance 是" << container[2]
                    << ", Distchng 是" << container[3] << ", Dirchng 是
" << container[4] << ";";
        }
        else if ((*itt).find("player") != (*itt).npos)
        {
            if ((*itt).find("hfut1") != (*itt).npos)
            {
                teammates.push_back(*itt);
            }
            else if ((*itt).find("hfut2") != (*itt).npos)
            {
                opponent.push_back((*itt));
            }

            cout << container[0] << " 距离我的 Direction
是 " << container[1] << ", Distance 是"
                    << container[2] << ", Dischng 是
" << container[3] << " Dirchng 是 " << container[4]
                            << " 它的 BodyDir 是 " << container[5] << ",
HeadDir 是 " << container[6] << ";";
```

```
        }
        else if ((*itt).find("goal") != (*itt).npos)
        {
            goal.push_back((*itt));

            cout << container[0] << " 距离我的 Direction 是
" << container[1] << " , Distance 是"
                << container[2] << ";";
        }
        else if ((*itt).find("f ") != (*itt).npos)
        {
            flag.push_back((*itt));

            cout << container[0] << " 距离我的 Direction
是 " << container[1] << " , Distance 是"
                    << container[2] << ";";
        }
        else if ((*itt).find("Line") != (*itt).npos)
        {
            Line.push_back((*itt));

            cout << container[0] << " 距离我的角度
是 " << container[1] << ";";
        }

        ++itt;
    }

}

message_manager::message_manager(string original_message)
{
    original_message.replace(original_message.size()-1, 1, " ");
    messages.append(original_message.replace(0, 1, " "));
    messages.replace(messages.find("see")-
2, 2, " ");                      //去左右两个括号

    //将字符串切割存到对应变量
    messages_hear.append(messages,messages.find("hear"), messages.find(
"see") - messages.find("hear"));
    messages_see.append(messages, messages.find("see"), messages.size()
 - messages.find("see"));

    int index = 0;
    int flag = 0;

    while (index < messages_hear.size())            //将听到的信息分
割
    {
        string temp = "";
        while (((messages_hear[index] != ' ') && (index < messages_hear
.size())) || (flag != 0))
        {
            if (messages_hear[index] == '(')
            {
                ++flag;
            }
            if (messages_hear[index] == ')')
            {
                --flag;
            }

            temp.append(messages_hear,index,1);
            ++index;
        }

        ++index;
        if (temp != "")
        {
            HEAR.push_back(temp);
        }
    }

    index = 0;
    flag = 0;
    while (index < messages_see.size())                         //将看到
的信息分割
    {
        string temp = "";
        while (((messages_see[index] != ' ') && (index < messages_see.s
ize())) || (flag != 0))
        {
            if (messages_see[index] == '(')
            {
                ++flag;
            }
            if (messages_see[index] == ')')
            {
                --flag;
            }

            temp.append(messages_see,index,1);
            ++index;
        }

        ++index;
        if (temp != "")
        {
            SEE.push_back(temp);
        }
    }

    vector<string>::iterator itt = HEAR.begin();             //将 HEAR 到
对应信息存储在对应成员变量里面
    while (itt != HEAR.end())
    {
        switch (itt - HEAR.begin()-1)
        {
        case 0:
            time.append((*itt));
            break;
        case 1:
            sender.append((*itt));
            break;
        case 2:
            message.append((*itt));
            break;
        }
        ++itt;
    }

}

message_manager::~message_manager()
{
}

int main()
{
    string s = "(hear 1022 -30 passto(23,24))(see 1022 ((ball) -
20 20 1 -2) ((player hfut1 2) 45 23 0.5 1 22 40 )
                        ((goal r) 12 20) ((Line r) -30))";
    message_manager s1(s);
    s1.message_print();

    return 0;
}
```