```
*-----------------------------Ubahn--------------------*


/*
 * File:   UBahn.cpp
 * Author: Rami Chaari
 *
 * Created on 27. Januar 2020, 10:40
 */
#define _CRT_SECURE_NO_WARNINGS
#include "UBahn.h"
#include <iostream>
#include <time.h>


using namespace std;

UBahn::UBahn() {

}


UBahn::~UBahn() {
}

void UBahn::drive(){
    cout << "U" << line << " driving" << endl;
}

void UBahn::stop() {
    cout << "U" << line << " stopping" << endl;
}

void UBahn::arrive() {
    time_t timer;
    tm *nun;
    timer = time(NULL);
    nun = localtime(&timer);
    cout <<"U"<< line << " arrived at " << nun->tm_mday << "/" <<nun-
>tm_mon +1<<"/"<<nun->tm_year + 1900 << " "<<nun->tm_hour<<":"<<nun-
>tm_min<<":"<<nun->tm_sec<<endl; ;
}

void UBahn::enter() {
 cout << "Der Passagier steigt ein" << endl;
}

void UBahn::leave() {
 cout << "Der Passagier steigt aus" << endl;
}

*------------------------- Ubahn.h -------------------------*
```

```
#ifndef UBAHN_H
#define UBAHN_H
#include "IZug.h"

class UBahn: public IZug {
private:
    int line;
public:
    UBahn();
    ~UBahn();
    void drive();
    void stop();
    void arrive();
    void enter();
    void leave();
};

#endif /* UBAHN_H */

*------------------------- UBahnFSM.cpp ------------------------*

#include "UBahnFSM.h"

#include <iostream>
using namespace std;

UBahnFSM::UBahnFSM(UBahn ubahn) {
 this->state = STOPPED;
 this->ubahn = ubahn;
}

UBahnFSM::~UBahnFSM() {

}

void UBahnFSM::evalEvents(UBahnEvent event) {
 UBahnState nextState;
 switch (state) {
 case STOPPED:
  if (event==START_DRIVING){
   nextState = DRIVING;
  } else if((event == PASSENGER_ENTER) || (event == PASSENGER_LEAVE)){
   nextState = STOPPED;
  }
  break;
 case CLEAN_UP:
  if (event == CLEAN_UP_DONE) {
   nextState = STOPPED;
  }
  break;
 case DRIVING:
  if (event == STOP) {
   nextState = STOPPED;
  }
  else if (event == PASSENGER_ENTER) {
   nextState = CLEAN_UP;
  }
  else if (event == PASSENGER_LEAVE) {
   nextState = ERROR;
  }
  break;
 case ERROR:
  break;
 }
 state = nextState;
}

void UBahnFSM::evalState(){
 switch (state) {
 case ERROR:
  cout << "Error!" << endl;
  break;
 case DRIVING:
  ubahn.drive();
  break;
 case CLEAN_UP:
  cout << "cleaning up" << endl;
  break;
 case STOPPED:
  ubahn.stop();
  ubahn.arrive();
  break;
 }
}

UBahnState UBahnFSM::getState() {
 return state;
}


*--------------------- UBahnFSM.h -------------------------*

#include "UBahn.h"

enum UBahnState {DRIVING, STOPPED, ERROR, CLEAN_UP};

enum UBahnEvent {START_DRIVING, STOP, PASSENGER_ENTER,
 PASSENGER_LEAVE, CLEAN_UP_DONE};


class UBahnFSM {
private:
 UBahnState state;
 UBahn ubahn;
public:
 UBahnFSM(UBahn);
 ~UBahnFSM();
 void evalEvents(UBahnEvent);
 void evalState();
 UBahnState getState();
};

*----------------------------------- Person.h ---------------*


#ifndef PERSON_H
#define PERSON_H

#include "IZug.h"

class Person {
public:
    Person();
    void enterTrain(IZug* zug);
    void leaveTrain();
private:
    IZug* zug;
};

#endif /* PERSON_H */

*----------------------- Person.cpp -----------------------*



#include "Person.h"

Person::Person() {
}

void Person::enterTrain(IZug* zug){
    this->zug = zug; zug->enter();
}

void Person::leaveTrain(){
    zug->leave();
}

*-------------------------- IZug --------------------------*


#ifndef IZUG_H
#define IZUG_H
enum ZugTyp {
 PUBLIC,
 TRANSPORT
};

class IZug {
```

```
public:
 virtual void enter() = 0;
 virtual void leave() = 0;
 virtual ~IZug() {};
protected:
 int passenger_count;
 ZugTyp type;
};



#endif /* IZUG_H */

*------------------------- main von Ubahn ------------------*

/*
 * File:   main.cpp
 * Author: Rami Chaari
 *
 * Created on 27. Januar 2020, 10:35
 */

#include <cstdlib>
#include "UBahn.h"
#include "IZug.h"
#include "Person.h"

using namespace std;


int main(int argc, char** argv) {
    UBahn ubahn;
    ubahn.arrive();
    Person person;
    IZug *zug;
    zug = &ubahn;
    person.enterTrain(zug);
    person.leaveTrain();

    return 0;
}
```

```
Error using dbstatus
Error: File: C:\PR2\UbahnRami.m Line: 1 Column: 1
Invalid use of operator.
```

Published with MATLAB® R2018b