|  |
| --- |
| **2019년 11월 12일 실습보고서** |
| **7조 : 송재원, 조윤직, 양석준, 박진영** |
| **실습자료1 : virtual 사용하여 ticket 클래스 제작** |
| **소스코드** |
| #include <iostream>  #include "TickteManager.h"  using namespace std;  int main()  {  cout << "7조 : 송재원, 조윤직,양석준, 박진영" << endl;  TickteManager tm;  tm.showTickets();  GeneralTicket\* tt = new GeneralTicket(20, 2000, true);  tm.ticketing(tt);  tm.ticketing(new GeneralTicket(21, 1000, false));  tm.ticketing(new GeneralTicket(22, 2000, true));  tm.ticketing(new AdvanceTicket(23, 1000, 30));  tm.ticketing(new AdvanceTicket(24, 1000, 30));  tm.showTickets();  tm.showAdvanceTicket();  tm.showGeneralTicket();  tm.showGeneral\_creditTicket();  return 0;  }  --------------------------------------------------------------------------------------------  #pragma once  #include "GeneralTicket.h"  #include "AdvanceTicket.h"  class TickteManager  {  private:  Ticket\* ticktes[10];  std::string m\_name;  int count;  public:  TickteManager();  ~TickteManager();  double getTotalprice();  void showTickets();  void showAdvanceTicket();  void showGeneralTicket();  void showGeneral\_creditTicket();  void ticketing(Ticket \* ticket);  void ticketing(Ticket & ticket);  };  -----------------------------------------------------------------------------------------------------------------------  #include "TickteManager.h"  TickteManager::TickteManager()  {  count = 0;  }  TickteManager::~TickteManager()  {  }  double TickteManager::getTotalprice()  {  double total = 0;  for (int i = 0; i < count; i++) {  total += ticktes[i]->getPrice();  }  return total;  }  void TickteManager::showTickets()  {  cout << "티켓 출력\n";  for (int i = 0; i < count; i++) {  ticktes[i]->show();  }  cout << "총 티켓 금액은 " << getTotalprice() << endl;  }  void TickteManager::showAdvanceTicket()  {  double ad\_total = 0;  int ad\_count = 0;  for (int i = 0; i < count; i++) {  if (typeid(\*ticktes[i]).name() == typeid(AdvanceTicket).name()) {  ad\_count++;  ad\_total += ticktes[i]->getPrice();    }  }  cout << "advancedTicket의 판매 갯수는 " << ad\_count << " 총 판매 금액은 " << ad\_total << "입니다" << endl;    }  void TickteManager::showGeneralTicket()  {  double ge\_total = 0;  int ge\_count = 0;  for (int i = 0; i < count; i++) {  if (typeid(\*ticktes[i]).name() == typeid(GeneralTicket).name()) {    ge\_count++;  ge\_total += ticktes[i]->getPrice();  }  }  cout << "GeneralTicket의 판매 갯수는 " << ge\_count << " 총 판매 금액은 " << ge\_total << "입니다" << endl;  }  void TickteManager::showGeneral\_creditTicket()  {  double ge\_total = 0;  int ge\_count = 0;  for (int i = 0; i < count; i++) {  if (typeid(\*ticktes[i]).name() == typeid(GeneralTicket).name()) {  GeneralTicket\* gt = (GeneralTicket\*)ticktes[i];  if (gt->getPayBtCredit()) {  ge\_count++;  ge\_total += ticktes[i]->getPrice();  }  }  }  cout << "GeneralTicket중 카드 결재의 판매 갯수는 " << ge\_count << " 총 판매 금액은 " << ge\_total << "입니다" << endl;  }  void TickteManager::ticketing(Ticket \* ticket)  {  ticktes[count++] = ticket;  }  void TickteManager::ticketing(Ticket& ticket)  {  ticktes[count++] = &ticket;  }  -----------------------------------------------------------------------------------------------------------------------  #pragma once  class Ticket  {  private:    protected:  int number;  double price;  public:  Ticket();  virtual ~Ticket();  Ticket(int number, double price);  Ticket(const Ticket& copy);  int getNumber();  virtual double getPrice() = 0;  void setPrice(double price);  virtual void show();  };  -----------------------------------------------------------------------------------------------------------------------  #include "Ticket.h"  #include<iostream>  Ticket::Ticket():Ticket(0,0) {  }  Ticket::~Ticket() {  std::cout << "Ticket 소멸자 실행 --티켓 번호 :" << number << std::endl;  }  Ticket::Ticket(int number, double price):number(number),price(price)  {    }  Ticket::Ticket(const Ticket& copy):Ticket(copy.number,copy.price) {    }  int Ticket::getNumber() {  return number;  }  void Ticket:: setPrice(double price) {  this->price = price;  }  void Ticket::show() {  std::cout << "티켓 번호는 " << getNumber() << " 티켓 가격은 " << getPrice() << std::endl;  }  -----------------------------------------------------------------------------------------------------------------------  #pragma once  #include "Ticket.h"  #include <iostream>  class GeneralTicket :  public Ticket  {  private:  bool payBycredit;  public:  GeneralTicket();  GeneralTicket(GeneralTicket& copy);  ~GeneralTicket();  GeneralTicket(int number,double price,bool paybycredit);  double getPrice();  bool getPayBtCredit();  void show();  };  #include "GeneralTicket.h"  using namespace std;  GeneralTicket::GeneralTicket() :GeneralTicket(0, 0, false)  {  }  GeneralTicket::GeneralTicket(GeneralTicket& copy):Ticket(copy),payBycredit(copy.payBycredit)  {  }  GeneralTicket::~GeneralTicket()  {  cout << "GeneralTicket 소멸자 실행 --티켓 번호 :" << number << endl;  }  GeneralTicket::GeneralTicket(int number, double price, bool paybycredit) :Ticket(number, price),payBycredit(paybycredit)  {  }  double GeneralTicket::getPrice()  {  return price\* (payBycredit ? 1.1 : 1);  }  bool GeneralTicket::getPayBtCredit()  {  return payBycredit;  }  void GeneralTicket::show()  {  cout << "일반 티켓 번호는 " << getNumber() << " 티켓 가격은 " << getPrice() <<" "<< (payBycredit ? "카드결제 " : "현금 결제") << endl;  }  -----------------------------------------------------------------------------------------------------------------------  #pragma once  #include "Ticket.h"  #include <iostream>  using namespace std;  class AdvanceTicket :public Ticket  {  private:  int advanceDays;  public:  AdvanceTicket();  AdvanceTicket(AdvanceTicket&copy);  ~AdvanceTicket();  AdvanceTicket(int number,double price,int advanceDays);  double getPrice();  int getAdvanceDays();  void show();  };  #include "AdvanceTicket.h"  using namespace std;  AdvanceTicket::AdvanceTicket() :AdvanceTicket(0,0,0)  {  }  AdvanceTicket::AdvanceTicket(AdvanceTicket& copy):Ticket(copy),advanceDays(copy.advanceDays)  {  }  AdvanceTicket::~AdvanceTicket()  {  cout << "AdvanceTicket 소멸자 실행 --티켓 번호 :" << number << endl;  }  AdvanceTicket::AdvanceTicket(int number, double price, int advanceDays):Ticket(number,price), advanceDays(advanceDays)  {  }  double AdvanceTicket::getPrice()  {  double dd;  if (advanceDays > 30)  dd = 0.6;  else if (advanceDays > 20)  dd = 0.7;  else if (advanceDays > 10)  dd = 0.8;  else if (advanceDays > 5)  dd = 0.9;  else  dd = 1;    return price \* dd;  }  int AdvanceTicket::getAdvanceDays()  {  return advanceDays;  }  void AdvanceTicket::show()  {  cout << "사전 티켓 번호는 " << getNumber() << " 티켓 가격은 " << getPrice() << endl;  } |
| **실행결과** |
|  |