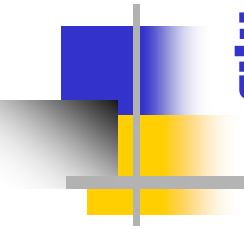

클래스 만들기 연습문제

프로그래밍 프로젝트

- Money 클래스 (두 가지 버전)
- 분수(fraction) 클래스
- Gas Pump 클래스

8,9번. Money 클래스

- Money 클래스와 테스트 프로그램 작성
  - Money 클래스로 금액이 몇 dollar 몇 cent인지를 표현
  - 금액의 dollar 부분만을 정수로 반환 (Getter 멤버 함수)
  - 금액의 cent 부분만을 정수로 반환 (Getter 멤버 함수)
  - 금액의 dollar 부분만을 지정 (Setter 멤버 함수)
  - 금액의 cent 부분만을 지정 (Setter 멤버 함수)
  - 금액 전체를 dollar로 환산하여 double 타입의 실수로 반환

- 1st Trial (8번)
  - 금액을 dollar와 cent를 분리해서 표현하는 방법

- 2nd Trial (9번)
  - 금액을 dollar와 cent를 합해서 하나로 표현하는 방법을 사용

8,9번. Money 클래스

```
int main ()
{
   Money m1, m2;

   m1.setDollars ( 20 );   // 20달러 35센트를 지정
   m1.setCents ( 35 );

   m2.setDollars ( 0 );  // 0달러 98센트를 지정
   m2.setCents ( 98 );

   cout << m1.getDollars () << "." << m1.getCents() << endl;  // 20.35를 출력
   cout << m1.getAmount() << endl;   // 20.35를 출력
   cout << m2.getAmount() << endl;   // 0.98을 출력
}
```

8,9번. Money 클래스 (1st Trial)

```
class Money
{
public:
        int getDollars ();  // 총 금액 중 달러에 해당하는 부분을 정수로 반환
        int getCents ();    // 총 금액 중 센트에 해당하는 부분을 정수로 반환
        void setDollars (int d); // 총 금액 중 달러에 해당하는 부분을 지정
        void setCents (int c); // 총 금액 중 센트에 해당하는 부분을 지정
        double getAmount (); // 총 금액을 달러로 환산한 실수 값을 반환
private:
        int dollars;        // 총 금액 중 달러에 해당하는 부분
        int cents;          // 총 금액 중 센트에 해당하는 부분
}
```

8,9번. Money 클래스 (1st Trial)

```
int Money::getDollars () { return dollars; } // 달러에 해당하는 부분만 반환
```

H1. getCents 멤버 함수 정의? // 센트에 해당하는 부분만 반환

H2. setDollars 멤버 함수 정의? // 주어진 정수로 달러를 지정

```
void Money::setCents ( int c ) { cents = c; } // 주어진 정수로 센트를 지정

double Money::getAmount ()  // 총 금액을 실수로 반환
{
    return static_cast<double>(dollars) +
                static_cast<double>(cents) / 100 ;
}
```

8,9번. Money 클래스 (2nd Trial)

```
// public 멤버들은 동일한 인터페이스를 갖고, private 멤버를 다르게 구현
class Money
{
public:
        int getDollars ();  // 총 금액 중 달러에 해당하는 부분을 정수로 반환
        int getCents ();    // 총 금액 중 센트에 해당하는 부분을 정수로 반환
        void setDollars (int d); // 총 금액 중 달러에 해당하는 부분을 지정
        void setCents (int c); // 총 금액 중 센트에 해당하는 부분을 지정
        double getAmount (); // 총 금액을 달러로 환산한 실수 값을 반환
private:
        double amount;      // 총 금액을 실수로 표현
}
```

8,9번. Money 클래스 (2nd Trial)

```
int Money::getDollars ()
{
   return                H3                ;  // 실수로 표현된 금액 중 소수점 이하 부분을
                                               // 버리면 dollars 금액
}

int Money::getCents ()
{
   int val =                    H4                    ; // 100을 곱한 값을 구한 다음

   return val % 100;          // 100으로 나눈 나머지를 구하여 센트를 계산
}
```

8,9번. Money 클래스 (2nd Trial)

```
void Money::setDollars (int d)
{
   int c = getCents ();   // 현재 금액에서 센트 부분 금액을 가져와 c에 저장
   amount =                          H5                          ; // 지정한 달러와 이전의 센트를
                                                                    // 합하여 총액을 계산
}

void Money::setCents (int c)
{
   int d =          H6          ;  // 현재 금액에서 달러를 저장
   amount = static_cast<double>( d ) + ( c / 100.0 );  // 이전의 달러와 지정한 센트를
                                                       // 합하여 총액을 계산
}

void Money::getAmount() {  return amount;  }   // 총 금액을 반환
```

5번. 분수 클래스

- 분수를 표현하는 클래스와 테스트 프로그램 작성
  - 두 개의 정수로 각각 분자와 분모를 표현
  - 분자를 설정하고 분모를 설정
  - 설정된 분수에 대한 double 타입의 실수 값을 반환
  - 설정된 분수를 약분해서 출력
    - 예: 20/60을 출력하는 대신 1/3을 출력

Fraction 클래스

- 분자: numerator
- 분모: denominator

```
class Fraction
{
public:
        double getDouble();                 // 실수로 환산환 분수의 값을 반환
        void outputReducedFraction();  // 주어진 분수를 약분해서 출력
        void setNumerator( int n );         // 분자를 설정
        void setDenominator ( int d );     // 분모를 설정
private:
        int numerator;        // 분자
        int denominator;     // 분모
        int gcd ();               // 분자와 분모의 최대 공약수를 찾는 멤버 함수
}
```

Fraction 사용

```
int main () {   // Fraction 클래스 테스트 프로그램

   Fraction f1, f2;

   f1.setNumerator ( 4 );                    // 분자로 4를 설정
   f1.setDenominator ( 2 );                  // 분모로 2를 설정
   cout << f1.getDouble () << endl;          // 해당 실수 값을 출력
   f1.outputReducedFraction ();              // 약분한 형태를 출력

   f2.setNumerator ( 20 );                   // 분자를 20, 분모를 60으로 설정하고
   f2.setDenominator ( 60 );
   cout << f2.getDouble () << endl;          // 해당 실수 값과
   f2.outputReducedFraction ();              // 약분한 형태를 출력
}
```

H7. 이 프로그램의 출력 결과?

Fraction 구현 [1/2]

```
void Fraction::setNumerator (int n) { numerator = n; }   // 분자를 설정

void Fraction::setDenominator (int d) { denominator = d; }   // 분모를 설정

double Fractioin::getDouble () {          // 실수 값으로 변환하여 반환
    return ( static_cast<double>(numerator) / denominator );
}

void Fraction::outputReducedFraction () {
    int g;

    g = gcd ();   // 분모와 분자의 최대 공약수를 구하고
    cout << numerator / g << " / " << denominator / g << endl;
                    // 분자와 분모를 최대 공약수로 나눈 값을 출력
}
```

Fraction 구현 [2/2]

```
int Fraction::gcd() {
    int g;

    // 분자와 분모 중 작은 수를 최대 공약수로 가정
    g = numerator > denominator ? denominator : numerator;

    // 가정한 최대 공약수가 1보다 크면
    while ( g > 1 ) {
        // 가정한 최대 공약수로 분자와 분모를 나눌 수 있다면 (나머지가 0이라면)
        if ( numerator % g  ==  0 && denominator % g   ==   0 )
          return g;    // 현재 가정한 수가 최대 공약수 이고
        g--;    // 어느 한 쪽이라도 나눌 수 없다면 현재 가정한 최대 공약수 보다
                //  1만큼 감소시킨 수를 최대 공약수로 가정
    }
    return 1;  // 1보다 큰 최대 공약수가 없다면 최대 공약수로 1을 반환
}
```

4번. “Gas Pump” 문제

- Gas Pump 클래스 및 테스트 프로그램 작성
  - 주유한 Gas 양을 표시
  - 주유한 Gas에 대한 금액을 표시
  - Gallon당 비용을 표시
  - 주유를 시작하기 전 주유한 Gas 양과 해당 금액을 0으로 초기화
  - 일단 Gas 주유를 시작하면 멈출 때까지 Gas 주유한 양과 해당 금액을 계속 관리
  - Gas 주유를 멈추기

Gas Pump 클래스

```
class GasPump
{
public:
        void initialize();                  // Gas Pump를 초기화
        void reset();                       // 주유를 시작하기 전 주유 양과 금액을 0으로
        void displayCostPerGallon();        // Gallon당 비용을 출력
        void displayGasNCharges();          // 주유한 Gas양과 금액을 출력
        void dispense();                    // Gas를 주유하면서 Gas 양과 금액을 업데이트
private:
        double gasDispensed, charge;  // 주유한 Gas 양, 해당 금액
public:
        void setPricePerGallon ( double newPrice);  // Gallon당 비용을 설정
        void buyFromJobber (double quantity);       // Gas Pump에 Gas를 채워넣음
        void displayAmountInMainTank ();            // Gas Pump에 담긴 Gas양을 출력
private:
        double gasInMainTank, costPerGallon; // Gas Pump에 담긴 Gas양, Gallon당 비용
}
```

Gas Pump 사용

```
int main () {   // GasPump 클래스 테스트 프로그램

   GasPump pump;

   pump.initialize();  // Gas Pump를 초기화
   pump.buyFromJobber(25);  // Gas Pump에 25갤론의 Gas를 채워넣음
   pump.setPricePerGallon (1.50); // Gas 가격을 갤론당 1.5달러로 지정

   pump.displayAmountInMainTank();   // Gas Pump에 담긴 전체 Gas양을 출력
   pump.dispense();  // Gas 주유를 시작

   pump.reset();        // Gas Pump의 주유한 Gas양과 가격을 0으로 초기화
   pump.displayAmountInMainTank(); // Gas Pump 남은 전체 Gas양을 출력
   pump.dispense();  // 다음 Gas 주유를 시작
}
```

Gas Pump 구현 [1/3]

```
void GasPump::initialize () {
    gasInMainTank = 0;     // Gas Pump에 담긴 Gas 양을 0으로 초기화
    gasDispensed = 0;      // 주유한 Gas양을 0으로 초기화
    charge = 0;            // 주유한 Gas양 가격을 0으로 초기화
}

void GasPump::reset () {
    H8. reset 메소드를 구현하시오.   // 이전에 주유한 Gas 양을 다시 0으로 초기화
                                    // 이전에 주유한 Gas에 대한 가격을 0으로 초기화
}

void GasPump::setPricePerGallon ( double newPrice ) {
    costPerGallon=newPrice;        // 새로운 Gas 가격 (Gallon당 달러)로 설정
}

void GasPump::buyFromJobber (double quantityBought) {
    gasInMainTank += quantityBought;   // Gas Pump에 quantityBought만큼
}                                      // Gas를 보충함
```

Gas Pump 구현 [2/3]

```
void GasPump::displayAmountInMainTank() {
    cout << gasInMainTank;
}

void GasPump::displayCostPerGallon () {
    cout <<
}

void GasPump::displayGasNCharges () {
     cout << "gallons: " << gasDispensed
          << "          $ " << charge << endl;
}
```

H9. 빈 칸을 채우시오

Gas Pump 구현 [3/3]

```
void GasPump::dispense()   {
   displayGasNCharges();                    // 현재 주유된 Gas양과 가격을 출력

   while (gasInMainTank > 0) {              // Gas Pump에 Gas가 있다면 주유를 진행
      char quit = cin.get();                // 문자 입력을 받아
      if ( quit == 'q' || quit 'Q' ) break; // 'q'거나 'Q'이면 주유를 중지
                                            // 그렇지 않으면 주유를 계속 진행
      gasDispensed += 0.1;                  // Gas 0.1 갤런을 주유
      charge += 0.1 * costPerGallon;        // 주유한 0.1 갤런의 Gas만큼 가격을 증가
         H10. 한 문장을 작성하시오          // 주유한 Gas만큼 Gas Pump의 Gas양을 감소

      displayGasNCharges();                 // 변경된 Gas 주유양과 가격을
   }
}
```