|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_14주 | 학번 : |  | 이름 : |  |

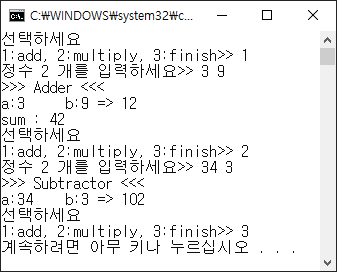
* **개념 적용 응용 프로그래밍 – 가상 함수와 추상클래스**

1. 다음과 같은 클래스를 상속받는 Adder, Mul 클래스를 구현하고 다형성을 사용하여 제시된 결과처럼 실행되는 프로그램을 완성 하시오. 단, 1을 선택하면 입력된 두 정수 a~b까지의 합을 계산하여 반환하는 sum() 함수를 호출하여 결과를 출력한다.

class Calculator {

protected:

int a, b;

public:

Calculator(int a, int b) : a(a), b(b) {}

virtual int calc() = 0;

virtual void write() {

cout << "a:" << a << “\tb:＂ << b <<“ =>＂ ;

}

};

class Adder : public Calculator {

public:

int sum(); //Adder 구현 시 추가

};

int main() {

Manage::run();

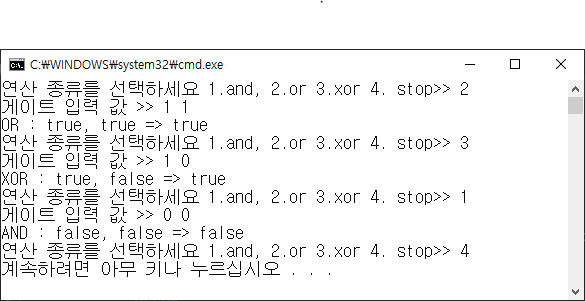
}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 두 개의 입력 신호를 받아 and, or, xor 연산을 수행한 결과를 출력하는 기본 게이트를 구현하는 프로그램을 완성하시오. 단, 각각의 게이트는 ANDGate, ORGate, XORGate 클래스로 작성하며 AbstractGate를 상속받는다. – 다형성을 적용할 것

* 논리형 상수 값 출력 시 아래 문장을 참고하세요

cout<<boolalpha<<true;

class AbstractGate {

public:

virtual bool operation(bool x, bool y) = 0; // 순수 가상 함수

};

int main() {

Manage::go();

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 주어진 Shape 클래스를 추상클래스로 만들고, Shape 추상 클래스를 상속받는 파생 클래스를 작성하여 제시된 결과처럼 실행될 수 있도록 프로그램을 완성하시오.

class Shape {

protected:

string name; // 도형의 이름

int width, height; // 도형이 내접하는 사각형

public:

Shape(string n = "", int w = 0, int h = 0) { name = n; width = w; height = h; }

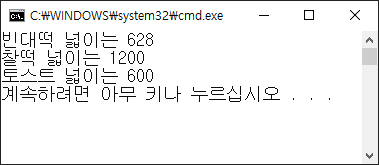
virtual double getArea() {

return 0;

}

string getName() { return name; } // 이름 리턴

};

int main() {

Shape \*p[3];

p[0] = new Oval("빈대떡", 10, 20);

p[1] = new Rect("찰떡", 30, 40);

p[2] = new Triangular("토스트", 30, 40);

for (int i = 0; i < 3; i++)

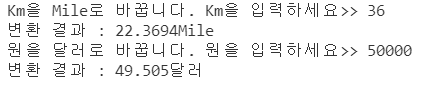
cout << p[i]->getName() << " 넓이는 " << p[i]->getArea() << endl;

for (int i = 0; i < 3; i++) delete p[i];

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 단위를 변환하는 Converter 클래스와 main()을 참고하여 프로그램을 완성하시오.



class Converter {

protected:

  double ratio;

  virtual double convert(double src)=0; // src를 다른 단위로 변환한다.

  virtual string getSourceString()=0; // 소스 단위 명칭

  virtual string getDestString()=0; // dest 단위 명칭

public:

  Converter(double ratio) { this->ratio = ratio; }

  void run() {

    double src;

    cout << getSourceString() << "을 " << getDestString() << "로 바꿉니다. ";

    cout << getSourceString() << "을 입력하세요>> ";

    cin >> src;

    cout << "변환 결과 : " << convert(src) << getDestString() << endl;

  }

};

int main() {

  KmToMile toMile(1.609344); // 1mile은 1.609344 Km

  WonToDollar wd(1010); // 1 달러에 1010원

  Converter \*cp = &toMile;

  cp->run();

  cp = &wd;

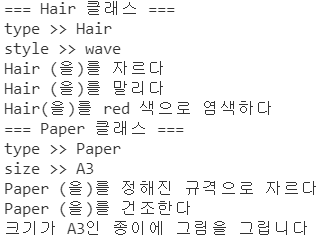
  cp->run();

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |

1. 제시된 코드와 실행 결과를 참조하여 프로그램을 완성 하시오.

class Tool{

  string type;

public:

  Tool() = default;

  Tool(string type) : type(type) {}

  string getType(){

    return type;

  }

  virtual void write(){

    cout << "type >> " << type << endl;

  }

  virtual void cut() = 0; //자르다

  virtual void dry() = 0; //말리다

};

1) Tool 클래스를 상속 받는 Hair 클래스 구현

* 헤어 스타일을 저장하는 필드 style 추가 , private
* 생성자 - 매개변수로 필드 초기화
* void tint(String color) – 기본 클래스 type을 color로 염색하는 함수 추가
* 기본클래스 write() 재정의 - type과 style 출력, 단, type은 기본클래스 write() 사용

2) Tool 클래스를 상속 받는 Paper 클래스 구현

* 용지 크기를 저장하는 필드 size 추가
* 생성자 - 매개변수로 필드 초기화
* void draw() – 기본 클래스 type에 그림을 그리는  메소드
* 기본 클래스 write() 재정의 - type과 size 출력, 단, type은 기본클래스 write() 사용

3) 전역함수 show 구현

int main(){

  Hair h("Hair", "wave");

  Paper p("Paper", "A3");

  show(h);

  show(p);

    return 0;

}

|  |
| --- |
| [프로그램 소스] |
| [실행 결과] |