|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2019\_1\_C++ \_11 | 학번 : | 20135151 | 이름 : | 이갑성 |

* **강의 내용**

|  |
| --- |
| 1. 추상 클래스와 순수 가상 함수 |
| #include <iostream>  using namespace std;  class Calculator { //추상 클래스  public:  virtual int add(int a, int b) = 0; // 두 정수의 합 리턴, 순수 가상 함수  virtual int subtract(int a, int b) = 0; // 두 정수의 차 리턴  virtual double average(int a [], int size) = 0; // 배열 a의 평균 리턴. size는 배열의 크기  };  class GoodCalc : public Calculator { //추상 클래스 구현  public:  int add(int a, int b) { return a + b; }  int subtract(int a, int b) { return a - b; }  double average(int a [], int size) {  double sum = 0;  for(int i=0; i<size; i++)  sum += a[i];  return sum/size;  }  };  int main() {  int a[] = {1,2,3,4,5};  Calculator \*p = new GoodCalc();  cout << p->add(2, 3) << endl;  cout << p->subtract(2, 3) << endl;  cout << p->average(a, 5) << endl;  delete p;  } |
| **[실행결과]** |

* 프로그램 과제

|  |
| --- |
| 1. 교재 467p : 문제 4번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class LoopAdder {  string name; //루프의 이름  int x, y, sum; //x에서 y까지의 합은 sum;  void read(); //x, y 값을 읽어 들이는 함수  void write(); // sum을 출력하는 함수  protected:  LoopAdder(string name = "")  {  this->name = name;  }  int getX() {  return x;  }  int getY() {  return y;  }  virtual int calculate() = 0; //순수 가상 함수. 루프를 돌며 합을 구하는 함수  public:  void run(); //연산을 진행하는 함수  };  void LoopAdder::read() {  cout << name << ":" << endl;  cout << "처음 수에서 두번째 수까지 더합니다. 두 수를 입력하세요 >> ";  cin >> x >> y;  }  void LoopAdder::write() {  cout << x << "에서 " << y << "까지의 합 = " << sum << " 입니다." << endl;  }  void LoopAdder::run() {  read();  sum = calculate();  write();  }  class WhileLoopAdder : public LoopAdder {  public:  WhileLoopAdder(string name) : LoopAdder(name) { }  int calculate() {  int result = 0;  int x = getX();  int y = getY();  while (true)  {  if (x > y)  {  break;  }  result += x;  x++;  }  return result;  }  };  class DoWhileLoopAdder : public LoopAdder {  public:  DoWhileLoopAdder(string name) :LoopAdder(name)  {  }  int calculate()  {  int result = 0;  int x = getX();  int y = getY();  do {  if (x > y)  {  break;  }  result += x;  x++;  } while (true);  return result;  }  };  int main()  {  WhileLoopAdder whileLoop("While Loop");  DoWhileLoopAdder doWhileLoop("Do While Loop");  whileLoop.run();  doWhileLoop.run();  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 467 : 문제5번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class AbstractGate {  protected:  bool x, y;  public:  void set(bool x, bool y) {  this->x = x;  this->y = y;  }  virtual bool operation() = 0;  };  class ANDGate : public AbstractGate {  public:  bool operation() {  bool result = true;  if (x == true && y == true)  result = true;  else if (x == false && y == false)  result = false;  else if (x == true && y == false)  result = false;  else if (x == false && y == true)  result = false;  return result;  }  };  class ORGate : public AbstractGate {  public:  bool operation() {  bool result;    if (x == true && y == true)  result = true;  else if (x == false && y == false)  result = false;  else if (x == true && y == false)  result = true;  else if (x == false && y == true)  result = true;  return result;  }  };  class XORGate : public AbstractGate {  public:  bool operation() {  bool result;  if (x == true && y == true)  result = false;  else if (x == false && y == false)  result = false;  else if (x == true && y == false)  result = true;  else if (x == false && y == true)  result = true;  return result;  }  };  int main()  {  ANDGate andGate;  ORGate orGate;  XORGate xorGate;  andGate.set(true, false);  orGate.set(true, false);  xorGate.set(true, false);  cout.setf(ios::boolalpha);  cout << andGate.operation() << endl;  cout << orGate.operation() << endl;  cout << xorGate.operation() << endl;  } |
| **[실행결과]** |

|  |
| --- |
| 1. 교재 469p : 문제7번 |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Shape {  protected:  string name;  int width, height;  public:  Shape(string n = "", int w = 0, int h = 0) {  name = n;  width = w;  height = h;  }  virtual double getArea() = 0;  string getName()  {  return name;  }  };  class Oval : public Shape { // 타원  public:  Oval(string n, int w, int h) :Shape(n, w, h) { }  double getArea()  {  return 3.14 \* width \* height;  }  };  class Rect : public Shape { // 사각형  public:  Rect(string n, int w, int h) : Shape(n, w, h) { }  double getArea()  {  return width \* height;  }  };  class Triangular : public Shape { // 삼각형  public:  Triangular(string n, int w, int h) : Shape(n, w, h) { }  double getArea()  {  return 0.5 \* width \* height;  }  };  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];  } |
| **[실행결과]** |