REPORT #5

|  |
| --- |
| **제 출 일** : 2017년 11월 21일 |
| **과 목 명** : 컴퓨터 프로그래밍 및 실습 |
| **담당교수 : 김정국 교수님** |
| **학 과** : 컴퓨터 전자 시스템 공학부 |
| **학 번** : 201702234 |
| **성 명 : 유동혁** |

**HANKUK UNIVERSITY OF FOREIGN STUDIES**

****

**1. 직원 / 프로그래머 / 매니저 프로그램**

* 1. **문제 정의**

-> 직원 클래스: 사원번호, 이름, 급여

프로그래머 클래스: 언어

매니저 클래스: 부서

**1.2 문제 해결 방법**

-> 프린트 함수를 하위 클래스에서도 사용하기 위해서 가상 함수를 이용

**1.3 소스 코드**

|  |
| --- |
| #include<iostream>  #include<string>  using namespace std;  class jick//최상위 클래스: 직원 클래스  {  protected://하위 클래스에 물려주는 변수  int num, won;//사원번호, 급여  string name;//이름  public:  jick();//직원 디폴트 생성자  jick(int n, string m, int w);//생성자  virtual void print();//직원 프린트: 가상함수를 사용해, 하위 클래스에서도 print라는 함수명을 사용할 수 있게 만듬  };  jick::jick()//직원 디폴트 생성자  {  num = 00000;//번호  name = "미설정";//이름  won = 0;//급여  }  jick::jick(int n, string m, int w)//직원 생성자  {  num = n;//번호  name = m;//이름  won = w;//급여  }  void jick::print()//직원 프린트  {  cout << "사원 번호: " << num << endl;  cout << "이름: " << name << endl;  cout << "급여: " << won << endl;  }  class prog : public jick//직원 클래스의 하위 클래스인 프로그래머 클래스  {  private:  string lang;//언어  public:  prog(int n, string m, int w, string l);//프로그래머  void print();//프로그래머 프린트  };  prog::prog(int n, string m, int w, string l)//프로그래머 생성자  {  num = n;//사원번호  name = m;//이름  won = w;//급여  lang = l;//언어  }  void prog::print()//프로그래머 프린트  {  cout << "사원 번호: " << num << endl;  cout << "이름: " << name << endl;  cout << "급여: " << won << endl;  cout << "언어: " << lang << endl;  }  class man : public jick//직원 클래스의 하위 클래스인 매니저 클래스  {  private:  string bu;//부서  public:  man(int n, string m, int w, string b);//매니저 생성자  void print();//매니저 프린트  };  man::man(int n, string m, int w, string b)//매니저 생성자  {  num = n;//사번  name = m;//이름  won = w;//급여  bu = b;//부서  }  void man::print()//매니저 프린트  {  cout << "사원 번호: " << num << endl;  cout << "이름: " << name << endl;  cout << "급여: " << won << endl;  cout << "부서: " << bu << endl;  }  int main()  {  jick a;//직원 a생성  jick b(12345, "김병호", 5000000);//직원 b생성  prog c(56789, "오순철", 4000000, "C언어");//프로그래머 c생성  man d(65432, "박금주", 4200000, "디자인팀");//매니저 d생성  a.print();  cout << "-------------------------" << endl;  b.print();  cout << "-------------------------" << endl;  c.print();  cout << "-------------------------" << endl;  d.print();  cout << "-------------------------" << endl;  return 0;  } |

**1.4 결과 화면**

|  |
| --- |
|  |

**2. 사람 / 강의자 / 학생 프로그램**

**2.1 문제 정의**

-> 사람 클래스: 이름, 생년월일

강의자 클래스: 강의 과목

학생 클래스: 전공

**2.2 문제 해결 방법**

-> 프린트 함수를 하위 클래스에서도 사용하기 위해 가상 함수를 이용

**2.3 소스 코드**

|  |
| --- |
| #include<iostream>  #include<string>  using namespace std;  class person//최상위 클래스 person  {  protected://하위 클래스에 물려주는 변수  string name;//이름  int birth;//생년월일  public:  person();//person 디폴트 생성자  person(string n, int b);//person 생성자  virtual void print();//가상함수를 이용해 print함수명을 하위 클래스에서도 사용가능하게 함  };  person::person()//person 디폴트 생성자  {  name = "미설정";//이름  birth = 20170000;//생년월일  }  person::person(string n, int b)//person 생성자  {  name = n;//이름  birth = b;//생년월일  }  void person::print()//person프린트  {  cout << "이름: " << name << endl;  cout << "생년월일: " << birth << endl;  }  class gang : public person//person클래스의 하위 클래스인 강의자 클래스  {  private:  string gwa;//강의과목  public:  gang(string n, int b, string g);//강의자 생성자  void print();//강의자 프린트  };  gang::gang(string n, int b, string g)//강의자 생성자  {  name = n;//이름  birth = b;//생년월일  gwa = g;//강의 과목  }  void gang::print()//강의자 프린트  {  cout << "이름: " << name << endl;  cout << "생년월일: " << birth << endl;  cout << "강의 과목: " << gwa << endl;  }  class stud : public person//person클래스의 하위 클래스인 학생 클래스  {  private:  string jeon;//전공 과목  public:  stud(string n, int b, string j);//학생 생성자  void print();//학생 프린트  };  stud::stud(string n , int b, string j)//학생 생성자  {  name = n;//이름  birth = b;//생년월일  jeon = j;//전공 과목  }  void stud::print()//학생 프린트  {  cout << "이름: " << name << endl;  cout << "생년월일: " << birth << endl;  cout << "전공: " << jeon << endl;  }  int main()  {  person a;//a 생성  person b("김병호", 19981114);//b 생성  gang c("오순철", 19851028, "C언어");//c 생성  stud d("박금주", 19981110, "컴퓨터 공학");//d 생성  a.print();  cout << "-------------------------" << endl;  b.print();  cout << "-------------------------" << endl;  c.print();  cout << "-------------------------" << endl;  d.print();  cout << "-------------------------" << endl;  return 0;  } |

**2.4 결과 화면**

|  |
| --- |
|  |

**3. 계좌 / 이자 / 적금 프로그램**

**3.1 문제 정의**

-> 계좌 클래스: 계좌번호, 잔액, 입금함수, 출금함수

이자 클래스: 이자율, 이자 계산 함수

적금 클래스: 적금 계좌번호, 적금 금액, 적금 잔액, 적금 계산 함수

**3.2 문제 해결 방법**

-> 이자 계산 함수: 이자 = 입력 받은 이자율 \* 현재 잔액 / 100

-> 적금 계산 함수: 잔액 = 기존 잔액 – 적금 금액

적금 잔액 = 기존 적금 잔액 + 적금 금액

**3.3 소스 코드**

|  |
| --- |
| #include<iostream>  #include<string>  using namespace std;  class gyae//최상위 클래스인 계좌클래스  {  protected://하위 클래스에 물려주는 변수  string num;//계좌번호  int jan;//잔액  public:  gyae();//계좌 디폴트 생성자  gyae(string n, int j);//계좌 생성자  void ip(int i);//입금함수  void chul(int c);//출금함수  virtual void print();//프린트 가상함수  };  gyae::gyae()//계좌 디폴트 생성자  {  num = "620-111222-333";  jan = 1000000;  }  gyae::gyae(string n, int j)//계좌 생성자  {  num = n;//계좌번호 설정  jan = j;//잔액 설정  }  void gyae::ip(int i)//입금함수  {  jan += i;  }  void gyae::chul(int c)//출금함수  {  jan -= c;  }  void gyae::print()//계좌 프린트 함수  {  cout << "계좌번호: " << num << endl;  cout << "잔액: " << jan << endl;  }  class yee : public gyae//계좌 클래스의 하위클래스: 이자 클래스  {  private:  int yul, ija;  public:  yee(string n, int j, int y);//이자 생성자  void print();//이자 프린트 함수  };  yee::yee(string n, int j, int y)//이자 생성자  {  num = n;//계좌 설정  jan = j;//잔액 설정  yul = y;//이자율 설정  ija = jan / 100 \* yul;//이자 설정  }  void yee::print()//이자 프린트 함수  {  cout << "계좌번호: " << num << endl;  cout << "잔액: " << jan << endl;  cout << "이자율: " << yul << endl;  cout << "이자 금액: " << ija << endl;  }  class juck : public gyae//계좌 클래스의 하위클래스: 적금 클래스  {  private:  string jucknum;//적금 계좌 번호  int juckjan, wonjan, juckack;//적금잔액, 원래 계좌 잔액, 적금금액  public:  juck(string jn, int jj);//적금 생성자  void juckfunc(int ja);//적금 함수  void print();//적금 프린트  };  juck::juck(string jn, int jj)//적금 생성자  {  jucknum = jn;//적금계좌  juckjan = jj;//적금잔액  wonjan = 1000000;//원래 계좌 잔액  juckack = 0;//적금금액  }  void juck::juckfunc(int ja)//적금함수  {  juckack = ja;  wonjan -= ja;  juckjan += ja;  }  void juck::print()//적금 프린트  {  cout << "계좌번호: 620-111222-333" << endl;  cout << "잔액: " << wonjan << endl << endl;  cout << "적금 계좌번호: " << jucknum << endl;  cout << "적금 금액: " << juckack << endl;  cout << "적금 잔액: " << juckjan << endl;  }  int main()  {  gyae a;//계좌 a생성  gyae b("620-123456-789", 1500000);//계좌 b생성  yee c("620-123456-789", 1000000, 3);//이자 c생성  juck d("620-987654-123", 0);//적금 d생성  b.ip(5000);//입금함수  b.chul(7000);//출금함수  d.juckfunc(200000);//적금함수  a.print();  cout << "----------------------" << endl;  b.print();  cout << "----------------------" << endl;  c.print();  cout << "----------------------" << endl;  d.print();  cout << "----------------------" << endl;  d.juckfunc(100000);//적금함수  d.print();  cout << "----------------------" << endl;  return 0;  } |

**3.4 결과 화면**

|  |
| --- |
|  |

**4. 근로자 / 시간 근로자 / 월급 근로자 프로그램**

* 1. **문제 정의**

-> 근로자 클래스: 이름, 시간당 임금

시간 근로자 클래스: 일한 시간, 주급 계산 함수

월급 근로자 클래스: 일한 시간, 일한 개월 수, 월급 계산 함수

**4.2 문제 해결 방법**

-> 주급 계산 함수: 만약 일한 시간이 40시간 이하면, 주급 = 시간 임금 \* 일한 시간 만약 일한 시간이 40시간을 초과하면, 주급 = 시간 임금 \* 40시간 + 시간 임금 \* 40시간을 초과한 시간 \* 1.5

-> 월급 계산 함수: 월급 = 시간 임금 \* 50시간 \* 일한 개월 수

**4.3 소스 코드**

|  |
| --- |
| /\*  \* 5-4.cpp  \*  \* Created on: 2017. 11. 21.  \* Author: Yoo DongHyuk  \*/  #include<iostream>  #include<string>  using namespace std;  class geun//최상위 클래스: 근로자 클래스  {  protected://하위 클래스에 물려주는 변수  string name;//이름  int im;//시간당 임금  public:  geun();//근로자 디폴트 생성자  geun(string n, int i);//근로자 생성자  virtual void print();//프린트 함수  };  geun::geun()//근로자 디폴트 생성자  {  name = "미설정";  im = 10000;  }  geun::geun(string n, int i)//근로자 생성자  {  name = n;//이름  im = i;//임금  }  void geun::print()//프린트 함수  {  cout << "이름: " << name << endl;  cout << "시간당 임금: " << im << endl;  }  class si : public geun//근로자 함수의 하위 클래스인 시간 근로자 클래스  {  private:  int work, jugeup;//일한시간, 계산된 주급  public:  si(string n, int i, int w);//시간근로자 생성자  void print();//시간근로자 프린트 함수  };  si::si(string n, int i, int w)//시간근로자 생성자  {  name = n;//이름  im = i;//임금  work = w;//일한 시간  if(work <= 40)//만약 일한 시간이 40시간 이하면  {  jugeup = im \* work;//시간당 임금 \* 일한 시간  }  else if(work > 40)//만약 40시간 초과면  {  jugeup = im\*40 + im\*(work-40)\*3/2;//시간당 임금 \* 40시간 + 시간당 임금 \* 초과시간 \* 1.5  }  }  void si::print()//시간근로자 프린트 함수  {  cout << "이름: " << name << endl;  cout << "시간당 임금: " << im << endl;  cout << "일한 시간: " << work << endl;  cout << "계산된 주급: " << jugeup << endl;  }  class wol : public geun//근로자 클래스의 하위 클래스인 월급근로자 클래스  {  private:  int work, month, wolgeup;//일한 시간, 일한 달 수, 계산된 월급  public:  wol(string n, int i, int w, int m);//월급근로자 생성자  void print();//월급근로자 프린트 함수  };  wol::wol(string n, int i, int w, int m)//월급근로자 생성자  {  name = n;//이름  im = i;//임금  work = w;//일한 시간  month = m;//일한 달 수  wolgeup = im \* month \* 50;//계산된 월급 = 임금 \* 달 수 \* 50시간  }  void wol::print()//월급근로자 프린트 함수  {  cout << "이름: " << name << endl;  cout << "시간당 임금: " << im << endl;  cout << "일한 시간: " << work << endl;  cout << "계산된 월급: " << wolgeup << endl;  }  int main()  {  geun a;//근로자 a생성  geun b("김병호", 10000);//근로자 b생성  si c("오순철", 15000, 30);//시간근로자 c생성  si d("김강호", 15000, 44);//시간근로자 d생성  wol e("박금주", 15000, 54, 3);//월급근로자 e생성  a.print();//a프린트  cout << "-----------------------" << endl;  b.print();//b프린트  cout << "-----------------------" << endl;  c.print();//c프린트  cout << "-----------------------" << endl;  d.print();//d프린트  cout << "-----------------------" << endl;  e.print();//e프린트  cout << "-----------------------" << endl;  return 0;  } |

**4.4 결과 화면**

|  |
| --- |
|  |