|  |
| --- |
| #include <iostream> |
| #include <iomanip> |
| using namespace std; |
| class Employee{ |
| private: |
| int id; |
| string name,gender,position; |
| double salary; |
| public: |
| Employee(){ |
| id=0; |
| name = "Unknow"; |
| gender ="Unknow"; |
| position = "Unknow"; |
| salary = 0; |
| } |
| int getId(){ |
| return id; |
| } |
| string getName(){ |
| return name; |
| } |
| string getGender(){ |
| return gender; |
| } |
| string getPosition(){ |
| return position; |
| } |
| double getSalary(){ |
| return salary; |
| } |
| void setId(int id){ |
| this->id=id; |
| } |
| void setName(string name){ |
| this->name=name; |
| } |
| void setGender(string gender){ |
| this->gender=gender; |
| } |
| void setPosistion(string position){ |
| this->position=position; |
| } |
| void setSalary(double salary){ |
| this->salary = salary; |
| } |
| void input(){ |
| cout << "Input Employee id      : "; cin >>id; |
| cout << "Input Employee name        : "; cin.clear();cin.ignore();getline(cin,name); //cin >>name |
| cout << "Input Employee gender      : "; cin >>gender; |
| cout << "Input Employee position        : "; cin >>position; |
| cout << "Input Employee salary      : "; cin >>salary; |
| } |
| void output(){ |
| cout << setw(10) <<id; |
| cout << setw(15) <<name; |
| cout << setw(10) <<gender; |
| cout << setw(15) <<position; |
| cout << setw(10) <<salary<<endl; |
| } |
| void printHeader(){ |
| cout << setw(10) <<"ID"; |
| cout << setw(15) <<"Name"; |
| cout << setw(10) <<"Gender"; |
| cout << setw(15) <<"Position"; |
| cout << setw(10) <<"Salary"<<endl; |
| } |
| void getList(Employee obj[],int n){ |
| printHeader(); //objE[0].printHheader() //objE->printHeader |
| for (int i = 0; i < n; i++){ |
| obj[i].output(); |
| } |
| } |
|  |
| //SearchByID |
| void searchById(Employee obj[], int n){ |
| int idSearch, isFound; |
| cout<<"Input Employee id to search : ";cin >>idSearch; |
| isFound=0; |
| for(int i=0; i<n; i++){ |
| if(idSearch == obj[i].getId()){ |
| printHeader(); |
| obj[i].output(); |
| isFound=1; |
| break; |
| } |
| } |
|  |
| if(isFound==0){ |
| cout<<"Message >>>(ID"<<idSearch<<"not fourn "<<endl; |
| } |
| } |
|  |
|  |
| //SearchByName |
| void searchByName(Employee obj[], int n){ |
| int isFound; |
| string textSearch; |
| cout<<"Input Employee name to search : ";cin>>textSearch; |
| isFound=0; |
| for(int i=0;i<n;i++){ |
| if(textSearch == obj[i].getName()){ |
| printHeader(); |
| obj[i].output(); |
| isFound=1; |
| break; |
| } |
| } |
| if(isFound == 0){ |
| cout<<"Message >>>(Name" <<textSearch<<"not fournd "<<endl; |
| } |
| } |
|  |
| //SearchByPosition |
| void searchByPosition(Employee obj[], int n){ |
| int isFound; |
| string textSearch; |
| cout<<"Input Employee name to search : ";cin>>textSearch; |
| isFound=0; |
| for(int i=0;i<n;i++){ |
| if(textSearch == obj[i].getPosition()){ |
| printHeader(); |
| obj[i].output(); |
| isFound=1; |
| break; |
| } |
| } |
| if(isFound == 0){ |
| cout<<"Message >>>(Position"<<textSearch<<"not fournd "<<endl; |
| } |
| } |
|  |
|  |
| void sortNameASC(Employee obj[],int n){ |
| Employee objTmp; |
| for (int i=0; i<n; i++){ |
| for(int j=i+1; j<n;j++){ |
| if(obj[i].getName() > obj[j].getName()){ |
| objTmp = obj[i]; |
| obj[i] = obj[j]; |
| obj[j] = objTmp; |
| } |
| } |
| } |
| cout<<"========== Sort name ASC =========="<< endl; |
| getList(obj,n); |
| } |
|  |
| //SortNmame |
| void sortNameDESC(Employee obj[],int n){ |
| Employee objTmp; |
| for(int i=0; i<n; i++){ |
| for (int j=i+1; j<n; j++){ |
| if(obj[i].getName()<obj[j].getName()){ |
| objTmp = obj[i]; |
| obj[i] = obj[j]; |
| obj[j] = objTmp; |
| } |
| } |
| } |
| cout<<"========== Sort name SESC =========="<< endl; |
| getList(obj,n); |
| } |
|  |
|  |
| void sortSalaryASC(Employee obj[],int n){ |
| Employee objTmp; |
| for(int i=0; i<n;i++){ |
| for(int j=i+1;j<n;j++){ |
| if(obj[i].getSalary()>obj[j].getSalary()){ |
| objTmp = obj[i]; |
| obj[i] = obj[j]; |
| obj[j] = objTmp; |
| } |
| } |
| } |
| cout<<"========== ort name SESC =========="<< endl; |
| getList(obj,n); |
| } |
|  |
|  |
| //SortSalary |
| void sortSalaryDESC(Employee obj[],int n){ |
| Employee objTmp; |
| for(int i=0; i<n;i++){ |
| for(int j=i+1;j<n;j++){ |
| if(obj[i].getSalary()<obj[j].getSalary()){ |
| objTmp = obj[i]; |
| obj[i] = obj[j]; |
| obj[j] = objTmp; |
| } |
| } |
| } |
| cout<<"========== Sort name SESC =========="<<endl; |
| getList(obj , n); |
| } |
|  |
|  |
| void deleteById(Employee obj[],int &n){ |
| int idSearch,isFound; |
| cout <<"Input eployee id to delete : "; cin >>idSearch; |
| isFound =0; |
| for (int i =0; i <n ; i++){ |
| if(idSearch == obj[i].getId()){ |
| for(int j=i; j<n; j++){ |
| obj[j] = obj[j+1]; |
| } |
| n--; |
| isFound =1; |
| break; |
| } |
| } |
| if(isFound == 0){ |
| cout<<"Message id << "<<idSearch <<" >>not found"<<endl; |
| } |
| getList(obj,n); |
| } |
|  |
|  |
| void updateById(Employee obj[] ,int n){ |
| int idSearch, isFound, optionUpdate; |
| cout << "Input eployee id to update : "; cin >>idSearch; |
| isFound =0; |
| for (int i=0;i<n;i++){ |
| if(idSearch == obj[i].getId()){ |
| obj->printHeader(); |
| obj[i].output(); |
| cout <<"=== What you an update (1-5) ? ==="<<endl; |
| cout << "1. Update Name"<<endl; |
| cout << "2. Update Gender"<<endl; |
| cout << "3. Update Position"<<endl; |
| cout << "4. Update Salary"<<endl; |
| cout << "5. Update all"<<endl; |
| cout <<"Enter Option our want (1-5) : "; cin>>optionUpdate; |
| switch(optionUpdate){ |
| case 1: |
| cout<< "Input new name :"; cin >> obj[i].name; |
| break; |
| case 2: |
| cout<< "Input new gender :"; cin >> obj[i].gender; |
| break; |
| case 3: |
| cout<< "Input new position :"; cin>>obj[i].position; |
| break; |
| case 4: |
| cout<< "Input new salary :"; cin>>obj[i].salary; |
| break; |
| case 5: |
| cout<< "Input new name  :"; cin>>obj[i].name; |
| cout<< "Input new gender :"; cin >> obj[i].gender; |
| cout<< "Input new position :"; cin>>obj[i].position; |
| cout<< "Input new salary :"; cin>>obj[i].salary; |
| break; |
| } |
| isFound =1; |
| break; |
| } |
| } |
| if (isFound == 0){ |
| cout <<"Message id << "<<idSearch<<">>> not found in lis !"<<endl; |
| } |
| } |
|  |
|  |
| double getTotalsalary(Employee obj[], int n){ |
| double total = 0; |
| for(int i=0; i<n; i++){ |
| total += obj[i].getSalary(); |
| } |
| return total; |
|  |
| } |
| int getTotalEmployee(int n){ |
| cout << "========== Total Employee ==========" << endl; |
| cout << "Total Employee : " << n << endl; |
| cout << " ========== Summary ==========" << endl; |
|  |
| } |
| }; |
| int main(){ |
| system("clear"); |
|  |
| Employee objE[10] ,objETm; |
| int option,optionLevel1 , n=0, i, idSearch , isFound; |
| do{ |
| cout <<" 1. Input Employee " <<endl; |
| cout <<" 2. List All Employee "<<endl; |
| cout <<" 3. Search Employee " <<endl; |
| cout <<" 4. Sort Employee "<<endl; |
| cout <<" 5. Ddelete Emplyee " <<endl; |
| cout <<" 6. Update Employee "<<endl; |
| cout <<" 7. Get Total Salary "<<endl; |
| cout <<" 8. Get Total Employee "<<endl; |
| cout <<" 9. Ext Program "<< endl; |
| cout <<" Input Choice (1-8) :"; cin >>option; |
| switch(option){ |
| case 1 : |
| objE[n].input(); |
| n++; |
| break; |
| case 2 : |
| objE->getList(objE,n); |
| break; |
| case 3 : |
| do{ |
| cout <<"=== Search Employee ===" <<endl; |
| cout <<"1. Searcch by Id " <<endl; |
| cout <<"2. Search by Name "<<endl; |
| cout <<"3. Search by Position "<<endl; |
| cout <<"4. Back"<<endl; |
| cout <<"Input Chooice (1-4) : "; cin >>optionLevel1; |
| switch(optionLevel1){ |
| case 1 : objE->searchById(objE,n);break; |
| case 2 : objE->searchByName(objE,n);break; |
| case 3 : objE->searchByPosition(objE,n);break; |
|  |
| } |
|  |
|  |
| }while(optionLevel1 !=4); |
| break; |
| case 4: |
| do{ |
| cout <<"=== Sort Employee ==="<<endl; |
| cout <<"1. sort name ASC"<<endl; // ASC small to Lange |
| cout <<"2. Sort name Desc"<<endl; // DESC  l to s |
| cout <<"3. Sort salary ASC"<<endl; |
| cout <<"4. short salary DESC"<<endl; |
| cout <<"5. Back"<<endl; |
| cout <<"Output Choice (1-5) : "; cin >>optionLevel1; |
| switch(optionLevel1){ |
| case 1 : objE->sortNameASC(objE,n);break; |
| case 2 : objE->sortNameDESC(objE,n);break; |
| case 3 : objE->sortSalaryASC(objE,n);break; |
| case 4 : objE->sortSalaryDESC(objE,n);break; |
| } |
|  |
|  |
| }while(optionLevel1 !=5); |
| break; |
| case 5 : |
| objE->deleteById(objE,n);break; |
| break; |
| case 6: |
| objE->updateById(objE,n); break; |
| break; |
| case 7: |
| cout << "================== Total Salary ==================" << endl; |
| cout << "Total Salary Of All Employees : $"; |
| //                  cout << fixed << setprecision(2); |
| cout << objE->getTotalsalary(objE, n); |
| cout << endl; |
| cout << "==================================================" << endl; |
| break; |
| case 8: |
| objE->getTotalEmployee(n); |
| break; |
| case 9: |
| exit(0);break; |
| } |
| }while (option !=7); |
|  |
|  |
|  |
|  |
| return 0; |
| } |

**🧠 ពន្យល់អំពី void getList(Employee obj[], int n)**

* **void** 👉 មានន័យថា function មិនបានបញ្ជូនតម្លៃត្រឡប់ក្រោយ។
* **getList(...)** 👉 ជាឈ្មោះ method ដែលត្រូវបានប្រើដើម្បីបង្ហាញបញ្ជីនិយោជិកទាំងអស់។
* **Employee obj[]** 👉 ជា parameter អារេ (array) នៃ object Employee ដែលផ្ទុកនិយោជិកជាច្រើន។
* **int n** 👉 ជាចំនួននិយោជិកសរុបក្នុងអារេ obj[]។

🧾 មុខងារ getList() ធ្វើអ្វី?

|  |
| --- |
| void getList(Employee obj[], int n){ |
| printHeader(); // បោះពុម្ពចំណងជើងក្បាលតារាង |
| for (int i = 0; i < n; i++){ |
| obj[i].output(); // បោះពុម្ពព័ត៌មាននិយោជិករៀងរាល់នាក់ |
| } |
| } |

**🧪 ឧទាហរណ៍ប្រើនៅក្នុង main():**

|  |
| --- |
| int main(){ |
| int n; |
| cout << "Enter number of employees: "; |
| cin >> n; |
|  |
| Employee emp[n]; |
| for(int i = 0; i < n; i++){ |
| cout << "\nEnter information for employee #" << (i + 1) << ":\n"; |
| emp[i].input(); |
| } |
|  |
| cout << "\nList of Employees:\n"; |
| emp[0].getList(emp, n); |
|  |
| return 0; |
| } |

**👉 នៅទីនេះ emp[0].getList(emp, n); គ្រាន់តែប្រើ object emp[0] ហៅ method getList() ដើម្បីបង្ហាញព៌តមានទាំងអស់។**

|  |
| --- |
| //SearchByID |
| void searchById(Employee obj[], int n){ |
| int idSearch, isFound; |
| cout<<"Input Employee id to search : ";cin >>idSearch; |
| isFound=0; |
| for(int i=0; i<n; i++){ |
| if(idSearch == obj[i].getId()){ |
| printHeader(); |
| obj[i].output(); |
| isFound=1; |
| break; |
| } |
| } |
|  |
| if(isFound==0){ |
| cout<<"Message >>>(ID"<<idSearch<<"not fourn "<<endl; |
| } |
| } |
|  |

**🔍 ពន្យល់ត្រង់បន្ទាត់**