



Môn học

Lập trình hướng đối tượng



Nội dung

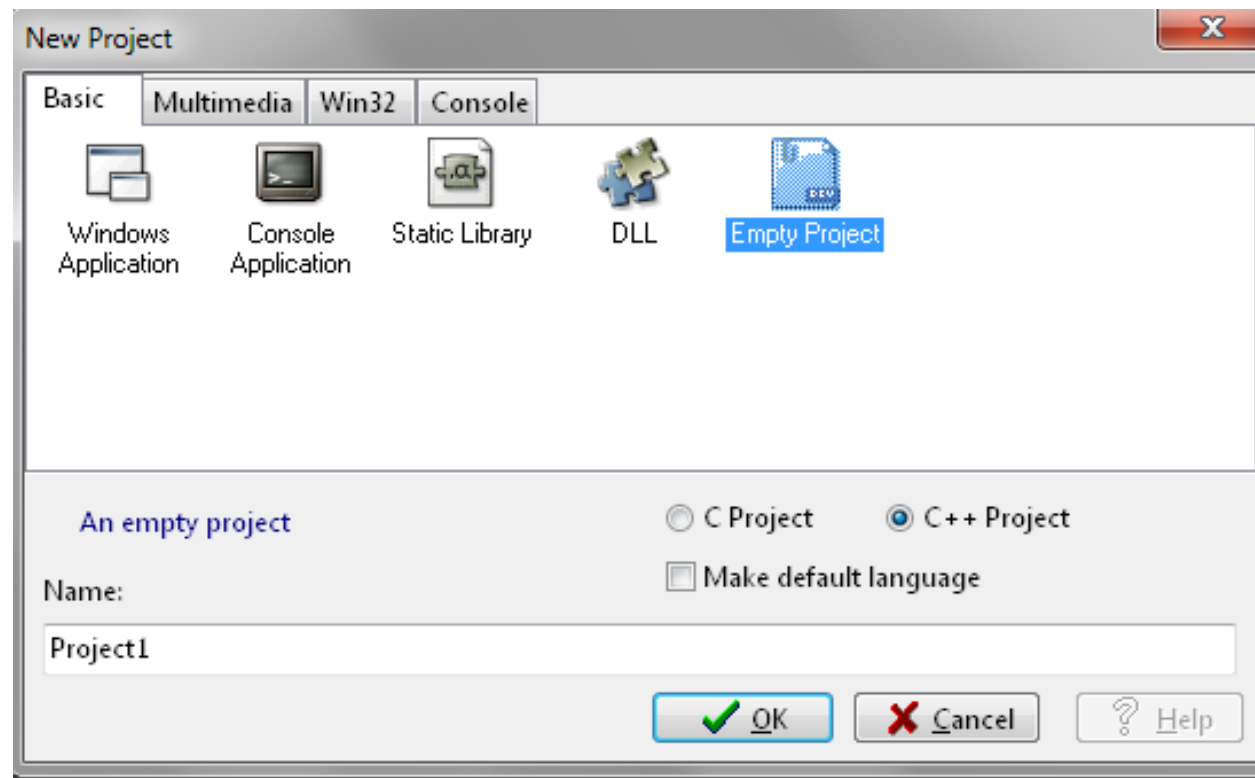
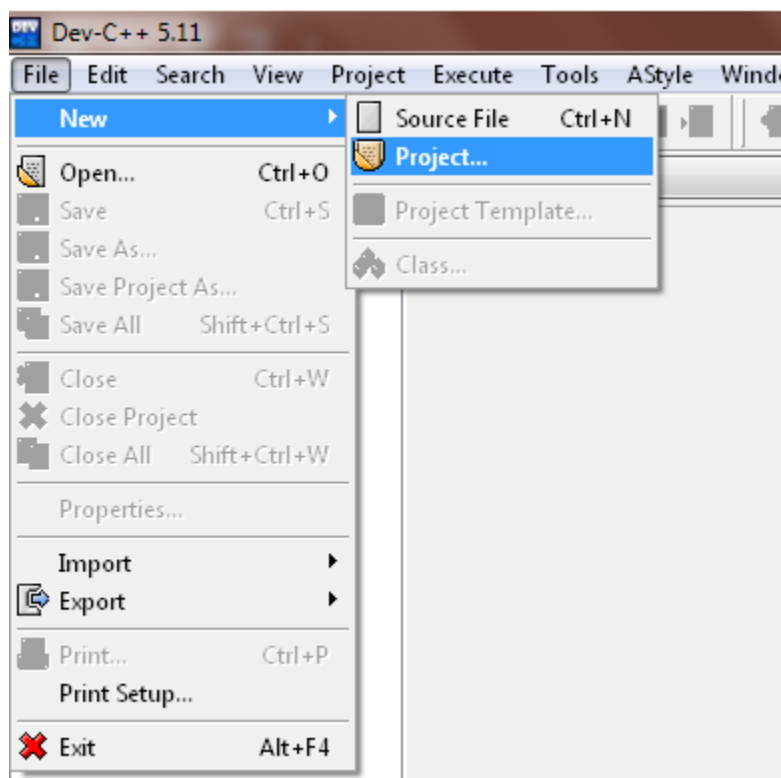
❖ Chương mở rộng: Thao tác với project trong Dev C++

- Hướng dẫn tạo project
- Ví dụ mẫu
- Bài tập



1. Hướng dẫn tạo Project

❖ File → New → Project → Empty Project → OK





1. Hướng dẫn tạo Project (t) [diem.h]

❖ Tạo file thư viện tự tạo “diem.h”

❖ #ifndef diem_h

❖ #define diem_h

===== Định nghĩa class tương ứng =====

❖ #endif



1. Hướng dẫn tạo Project (t) [diem.h]

===== Định nghĩa class tương ứng =====

```
#include <iostream>
#include <math.h>
using namespace std;
class Diem {
    private:
        double x, y;
    public:
        Diem(double x = 0, double y = 0);
        void dichuyen(double x, double y);
};
```



1. Hướng dẫn tạo Project (t) [diem.h]

===== Định nghĩa class tương ứng =====

public:

```
double kcach(Diem u);
```

```
friend istream& operator>>(istream &is, Diem &u);
```

```
friend ostream& operator<<(ostream &os, Diem u);
```

```
friend Diem operator+(Diem u, Diem v); // d3 = d1 + d2;
```

```
void operator+=(Diem u); // d1 += d2;
```



1. Hướng dẫn tạo Project (t) [Diem.cpp]

```
#include "diem.h"
Diem::Diem(double x, double y){
    this->x = x;
    this->y = y;
}
void Diem::dichuyen(double x, double y){
    this->x += x;
    this->y += y;
}
```



1. Hướng dẫn tạo Project (t) [Diem.cpp]

```
void Diem::operator+=(Diem u){  
    this->x += u.x;  
    this->y += u.y;  
}  
  
double Diem::kcach(Diem u){  
    return sqrt(pow(this->x - u.x, 2) + pow(this->y - u.y, 2));  
}
```




1. Hướng dẫn tạo Project (t) [Diem.cpp]

```
istream& operator>>(istream &is, Diem &u){  
    is >> u.x >> u.y;  
    return is;  
}  
ostream& operator<<(ostream &os, Diem u){  
    os << u.x << " " << u.y << endl;  
    return os;  
}
```

```
Diem operator+(Diem u, Diem v){  
    Diem res;  
    res.x = u.x + v.x;  
    res.y = u.y + v.y;  
    return res;  
}
```

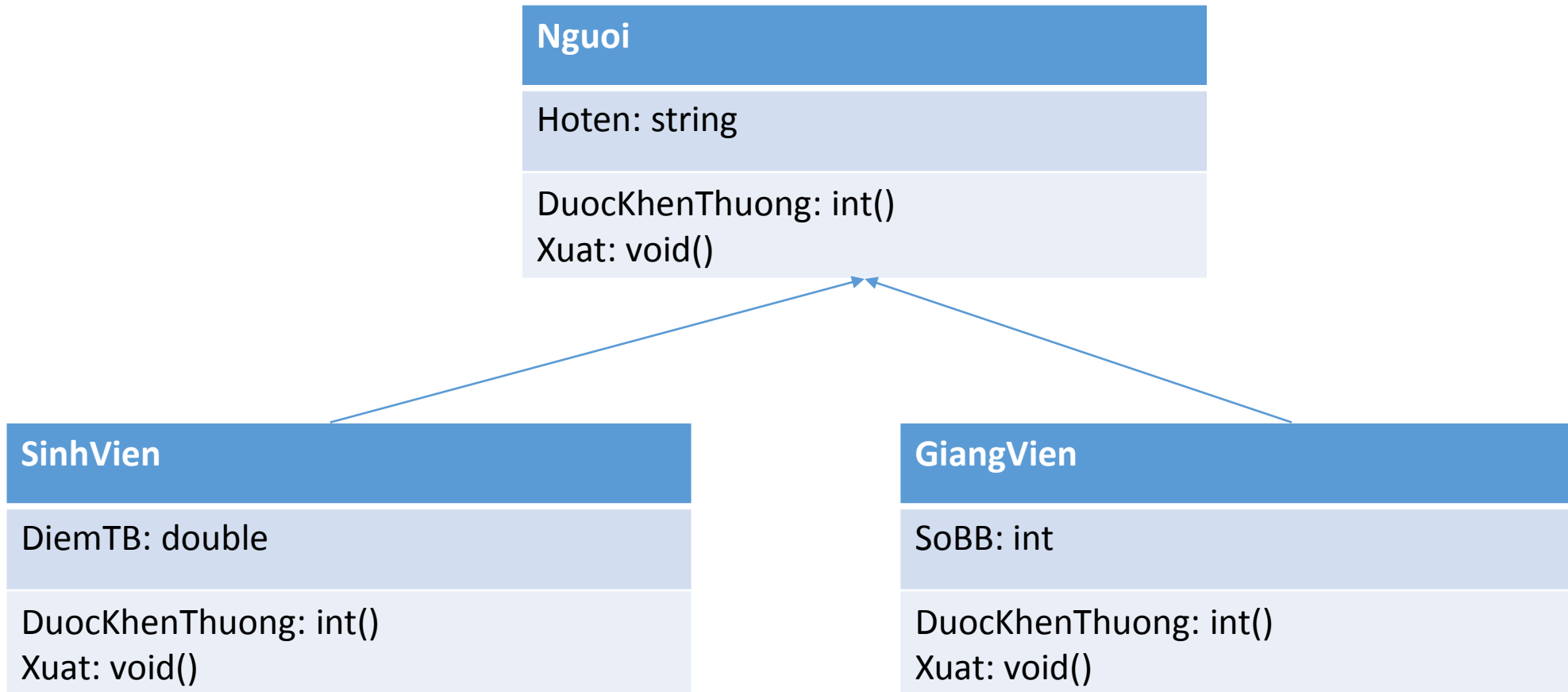


1. Hướng dẫn tạo Project (t) [Main.cpp]

```
#include "diem.h"
int main(){
    Diem d1; cin >> d1; cout << d1;
    Diem d2(3,4); cout << d1.kcach(d2) << endl;
    Diem d3 = d1 + d2; cout << d3;
    d1 += d2; cout << d1;
    return 0;
}
```



2. Ví dụ mẫu





2. Ví dụ mẫu (t) [PERSON.h]

```
#ifndef PERSON_H
#define PERSON_H
#include <iostream>
using namespace std;
class Nguoi {
    protected:
        string hoten;
    public:
        Nguoi(string hoten);
        virtual int DuocKhenThuong() const = 0;
        virtual void Xuat() const = 0;
};
#endif
```



2. Ví dụ mẫu (t) [STUDENT.h]

```
#ifndef STUDENT_H
#define STUDENT_H
#include "PERSON.h" // user tu tao thi dung "", neu thu vien co san thi dung <>
class SinhVien : public Nguoi { // ke thua public
    protected:
        double diemTB;
    public:
        SinhVien(string ht = "unknow", double diem = 0);
        virtual int DuocKhenThuong() const;
        virtual void Xuat() const;
};
#endif
```




2. Ví dụ mẫu (t) [TEACHER.h]

```
#ifndef TEACHER_H
#define TEACHER_H
#include "PERSON.h"
class GiangVien : public Nguoi { // kế thừa public
protected:
    int SoBaiBao;
public:
    GiangVien(string hoten="unknow", int sbb = 0);
    virtual int DuocKhenThuong() const;
    virtual void Xuat() const;
};
#endif
```



2. Ví dụ mẫu (t) [PERSON.cpp]

```
// Thực thi các phương thức của class Nguoi
#include "PERSON.h"
Nguoi::Nguoi(string hoten){
    this->hoten = hoten;
}
```



2. Ví dụ mẫu (t) [STUDENT.cpp]

```
// Thực thi các phương thức của class STUDENT
#include "STUDENT.h"
SinhVien::SinhVien(string hoten, double diem) : Nguoi(hoten){
    this->diemTB = diem;
}
int SinhVien::DuocKhenThuong() const {
    return this->diemTB >= 9;
}
void SinhVien::Xuat() const {
    cout << "hoten sinhvien: " << hoten << endl;
}
```



2. Ví dụ mẫu (t) [TEACHER.cpp]

```
// Thực thi các phương thức của class TEACHER
#include "TEACHER.h"
GiangVien::GiangVien(string hoten, int sbb) : Nguoi(hoten){
    this->SoBaiBao = sbb;
}
int GiangVien::DuocKhenThuong() const {
    return SoBaiBao >= 5;
}
void GiangVien::Xuat() const {
    cout << "hoten giangvien: " << hoten << endl;
}
```



2. Ví dụ mẫu (t) [Main.cpp]

```
#include "PERSON.h"  
#include "TEACHER.h"  
#include "STUDENT.h"  
#define MAXN 100
```

```
int main(){  
    Nguoi *dsNguoi[MAXN];  
    int cnt = 0;  
    do {  
        string hoten; char type;  
        cout << "ho ten: ";  
        fflush(stdin);  
        getline(cin, hoten);  
        cout << "GV hay SV: (G, S)?: ";  
        cin >> type;
```




2. Ví dụ mẫu (t) [Main.cpp]

```
if(type == 'G' || type == 'g'){
    int sbb; cout << "so bai bao: "; cin >> sbb;
    dsNguoi[cnt++] = new GiangVien(hoten, sbb);
}
else {
    double dtb;
    cout << "diem trung binh: "; cin >> dtb;
    dsNguoi[cnt++] = new SinhVien(hoten, dtb);
}
```



2. Ví dụ mẫu (t) [Main.cpp]

```
char chon; cout << "Tiep tục không? (Y/N): "; cin >> chon;
if(chon == 'N' || chon == 'n' || cnt >= MAXN) break;
} while(true);

cout << "=====" << endl;
for(int i = 0; i < cnt; i++){
    dsNguoi[i]->Xuat();
    if(dsNguoi[i]->DuocKhenThuong()){
        cout << "--> Duoc khen thưởng." << endl;
    }
}
```



3. Bài tập

- ❖ Bài tập 1: Xây dựng class quản lý Số phức (Complex) thực hiện các thao tác: hàm dựng, cộng trừ, nhân, chia và định nghĩa các phép toán nhập xuất.
- ❖ Bài tập 2: Xây dựng class Date thực hiện các thao tác: nhập xuất, hàm dựng, cho phép biết ngày trong tháng, năm nhuận,...
- ❖ Bài tập 3: Xây dựng class quản lý đối tượng đa thức và thực hiện các phép toán trên đa thức.
- ❖ Bài tập 4: Xây dựng class quản lý đối tượng ma trận và thực hiện các thao tác trên ma trận.



Thank you!

Any questions?