01.09.2010

Friend Class

Bastian Graser

dkfz.
DEUTSCHES KREBSFORSCHUNGSZENTRUM
IN DER HELMHOLTZ-GEMEINSCHAFT

Friend Klassen

Splitting one big class in two small ones

- Many commonly used members
- How to proceed?

  - A) Make everything „public“
    - Small effort
    - Encapsulation destroyed
  - B) Make „get“ and „set“ functions
    - Medium effort
    - Encapsulation destroyed (like A, just different name)
  - C) Complete refactoring of both classes
    - High effort

- D) Declare as friend classes
  - Small effort
  - Encapsulation preserved

Friend Klassen

```
class A
{
  private:
    int topSecret;
};

class B
{
  public:
    void change(A);
};

void B::change(A myParam) { myParam.topSecret = 5; }
```

Friend Klassen

```
class A
{
  private:
    int topSecret;
};

class B
{
  public:
    void change(A);
};

void B::change(A myParam) { myParam.topSecret = 5; }
```

Friend Klassen

```
class A
{
friend class B;
  private:
    int topSecret;
};

class B
{
  public:
    void change(A);
};

void B::change(A myParam) { myParam.topSecret = 5; }
```

Friend Klassen

```
class A
{
private:
    int topSecret;
     friend void change(A);
};

void change(A a) { a.topSecret = 5; }
```

Friend Klassen

- „Do friends violate encapsulation?“
  - „No! They *enhance* encapsulation“
- When using correctly
  - Easy method to split classes
  - Code remains readable and slim
  - Encapsulation preserved

Friend Klassen

- Zu Beachten:
- Friendship wird nicht vererbt
- Friendship beruht nicht auf Gegenseitigkeit
- Friendship wird nicht den Freunden meiner Freunde gewährt

Friend Klassen

- Keep in mind:
- Friendship is not inherited
- Friendship is always one-way
- Friendship is not declared to my friends friends

Workaround:
- add more friend declarations (maybe unpretty)

Friend Klassen

- Friendship is not inherited
  - Nice Workaround: **Virtual Friend Function Idiom**.

```
class A  {
friend class B;
private:
   void foo();
};

class B {
protected:
   void foo(A& a) {a.foo();}
   virtual void bar() = 0;
};

class C : public B {
private:
   virtual void bar() {  A a;  foo(a); }
};
```