

5.1. Защищенное наследование. Логика видимости и доступности родителей и прародителей, их членов при двухуровневом наследовании с различными модификаторами доступа. Логика запрета и разрешения обращений к членам родителей при использовании слова `friend` в разных местах с различными комбинациями приватного и защищенного наследования.

Защищенное наследование: Есть три типа наследования

1. `public`: получить доступ к родительской части можно из любого места программы
2. `protected`: получить доступ к родительской части можно только изнутри класса, из его наследников и друзей
3. `private`: получить доступ к родительской части можно только из тела класса и из его друзей

При многоуровневом наследовании подниматься (получать доступ) по `public` и `protected` наследованиям можно сколько угодно, но если у нас на пути встретилось `private` наследование, то этого предка и то что выше него мы не видим (если мы не непосредственный ребенок). Если мы ребенок `private` наследования, то мы можем получить доступ к родителю и подниматься дальше по `public` и `protected` (до следующего `private`).

Friends: Рассмотрим пример

```
1 struct Grandma {           struct Mom: protected Grandma {
2     int g = 0;              int m = 0;
3 };                          };
4
5 struct Daughter: private Mom {
6     int d = 0;
7     friend int main();
8 };
9
10 int main() {
11     Daughter b;
12     std::cout << b.g;
13 }
```

В этом случае все будет хорошо и `g` будет доступно из `main` (для `protected` в обоих местах аналогично). Однако если мы поменяем `private` и `protected` местами мы получим СЕ. Это интуитивно понятно: `Grandma` приватна для `Mother`, значит недоступна из ее наследников, а значит и из друзей наследников (очевидно что для двух `private` ситуация может только ухудшиться и доступа все еще не будет)