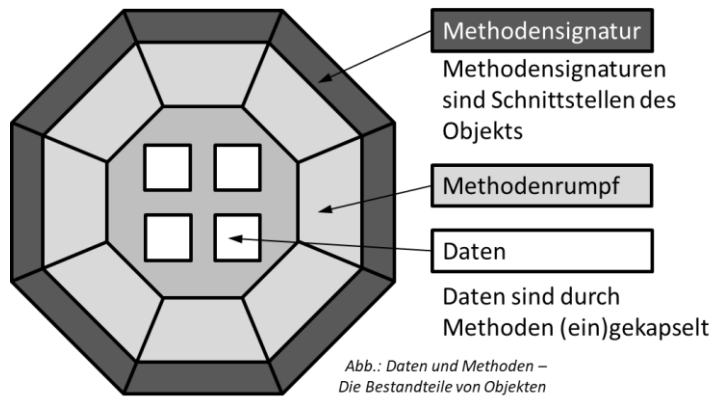




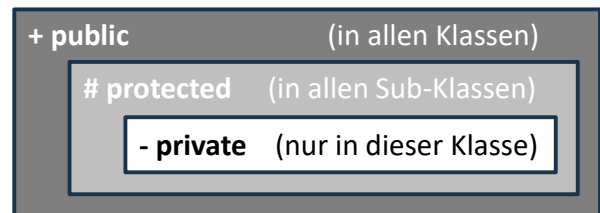
In der Informatik ist Datenkapselung ein Konzept, das in der objektorientierten Programmierung verwendet wird, um die Daten eines Objekts vor unberechtigtem Zugriff zu schützen. Dies wird erreicht, indem die Daten in privaten oder geschützten Feldern gespeichert werden, die nur von Methoden der Klasse selbst oder von abgeleiteten Klassen abgerufen werden können.



Datenkapselung hat mehrere Vorteile. Erstens hilft sie dabei, die Integrität der Daten zu schützen, indem verhindert wird, dass sie von unautorisierten Benutzern geändert werden. Zweitens kann sie dazu beitragen, den Code zu vereinfachen, indem die Anzahl der Methoden, die auf die Daten zugreifen müssen, verringert wird. Drittens kann sie dazu beitragen, den Code zu testen, indem es einfacher wird, die Auswirkungen von Änderungen an den Daten zu isolieren.

Sichtbarkeit (Zugriffsmodifizierer)

Diese Konzepte werden in C++ u.a. durch **Getter-** und **Setter-**Methoden mit dem entsprechenden Zugriffsmodifizierer (Darstellung der Sichtbarkeit) erzeugt.



```

Beispielcode der Klasse Zeit
① class Zeit {
②     private :
③         int stunde;
④         int minute;
⑤     protected :
⑥         void zeitReset() {
⑦             this->stunde = 0;
⑧             this->minute = 0; }
⑨     public :
⑩         Zeit(int h , int m) {
⑪             this->stunde = h;
⑫             this->minute =m; }
⑬         int getMinute() {
⑭             return minute; }
⑮         void setMinute(int m) {
⑯             this->minute=m; }
};
Zeit.CPP
    
```

Die Attribute (Instanzvariablen) sind nur innerhalb der Klasse sichtbar

Service-Methode ist nur in der Klasse Zeit und alle Unterklassen sichtbar und zugreifbar

(überladener) Konstruktor: in allen Klassen können Zeit-Objekte instanziiert werden.

Getter-Methode der Instanz-Variable minute von allen Klassen zugreifbar

Setter-Methode der Instanz-Variable minute von allen Klassen zugreifbar

