

Программирование на языке Си++

Модуль 2. ИНКАПСУЛЯЦИЯ

- Декларация класса
- Управление доступом
- Указатель `this`. Дружественные классы и функции

Понятие класса (начало)

- **Класс** — составной производный тип, *теоретически* объединяющий набор поименованных компонентов — данных (**членов данных, полей**) и функций их обработки (**методов, функций-членов**). *Практически* компонентами класса в языке Си++ могут быть данные, функции, классы, перечисления, битовые поля, дружественные функции, дружественные классы и имена типов [Под03]
- Класс — ключевое понятие парадигмы **объектно-ориентированного программирования** (ООП), реализующей принципы **инкапсуляции** данных и методов, **наследования** дочерними классами характеристик классов-родителей и **полиморфизма** в поведении базовых классов
- Переменная типа «класс» носит название **объекта**

Описание и определение классов и экземпляров (объектов)

- **(Опережающее) описание класса**

`class <имя типа> ;`

- **Определение класса**

`class <имя типа> { <определение компонентов> } ;`

- **Простейшее определение экземпляра (объекта)**

`<имя типа> <идентификатор> ;`

- **Определение указателя на экземпляр (объект) класса**

`<имя типа> * <идентификатор> ;`

- **Определение ссылки на экземпляр (объект) класса**

`<имя типа> &<идентификатор> ;`

Доступ к компонентам объекта

- **Доступ извне к (общедоступным нестатическим) компонентам экземпляра (объекта) через идентификатор объекта класса**
 - с использованием квалифицированного имени
<идентификатор> . <имя типа> :: <имя компонента>
 - с использованием уточненного имени
<идентификатор> . <имя компонента>
- **Доступ извне к (общедоступным нестатическим) компонентам экземпляра (объекта) через указатель на объект класса**
 - с использованием квалифицированного имени
<идентификатор> -> <имя типа> :: <имя компонента>
 - с использованием уточненного имени
<идентификатор> -> <имя компонента>

Спецификаторы доступа. Указатель `this`

- Для регулирования прав доступа к компонентам классов, а также объектов классов используются **спецификаторы доступа**:
 - `private` — для собственных компонентов;
 - `public` — для общедоступных компонентов;
 - `protected` — для защищенных компонентов
- **Указатель `this`** — константный указатель на объект класса, автоматически формируемый и неявно передаваемый в каждую (нестатическую) функцию – член класса при ее вызове как скрытый параметр. Определить или описать указатель `this` явным образом невозможно

Дружественные функции

- **Дружественная функция** — функция, которая, не будучи компонентом класса, имеет доступ к его защищенным и собственным компонентам [Под03]
- Особенности дружественных функций в языке Си++:
 - при вызове не получают указателя `this`;
 - не подчиняются спецификаторам доступа того класса, по отношению к которому они являются дружественными;
 - не могут быть компонентными функциями того класса, по отношению к которому они являются дружественными, однако могут быть компонентными функциями других классов;
 - могут быть дружественными по отношению к нескольким классам

- **Описание дружественной функции**

```
friend <тип результата> <идентификатор>  
    ([<спецификация формальных параметров>]) ;
```


Дружественные классы

- **Дружественный класс** — класс, все компонентные функции которого являются дружественными по отношению к другому классу
- Дружественный класс должен быть определен вне класса, по отношению к которому он является дружественным, и может быть определен после (ниже) него по тексту
- **Описание дружественного класса**

```
friend class <ИМЯ ТИПА> ;
```


Список литературы

- [КР92] Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ. — М.: Финансы и статистика, 1992. — 272 с.
- [КР06] Керниган Б., Ритчи Д. Язык программирования С / Пер. с англ. — М.: Вильямс, 2006. — 304 с.
- [Под03] Подбельский В.В. Язык Си++: Учеб. пособие. — 5-е изд. — М.: Финансы и статистика, 2003. — 560 с., ил.
- [Под04] Подбельский В.В., Фомин С.С. Программирование на языке Си. — 2-е доп. изд. — М., Финансы и статистика, 2004. — 600 с.