

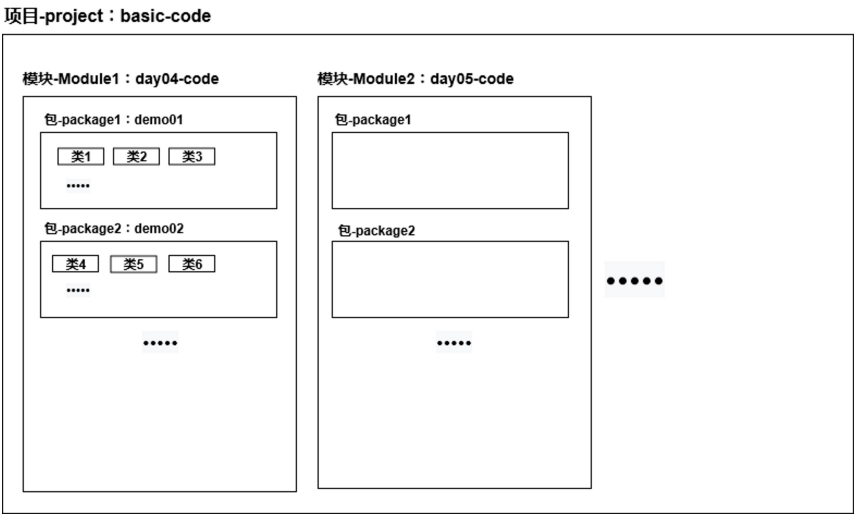
day04_IDEA、方法

一.开发工具IntelliJ IDEA

集成开发环境(IDE,Integrated Development Environment)是用于提供程序开发环境的应用程序，一般包括代码编辑器、编译器、调试器和图形用户界面等工具。集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件服务套。所有具备这一特性的软件或者软件套(组)都可以叫集成开发环境。如微软的Visual Studio系列，Borland的C++ Builder、Delphi系列等。该程序可以独立运行，也可以和其它程序并用。

IntelliJ IDEA是一个专门针对Java开发的集成开发工具(IDE)，由Java语言编写。所以，需要有JRE运行环境并配置好环境变量。它可以极大地提升我们的开发效率，可以自动编译，检查错误等。

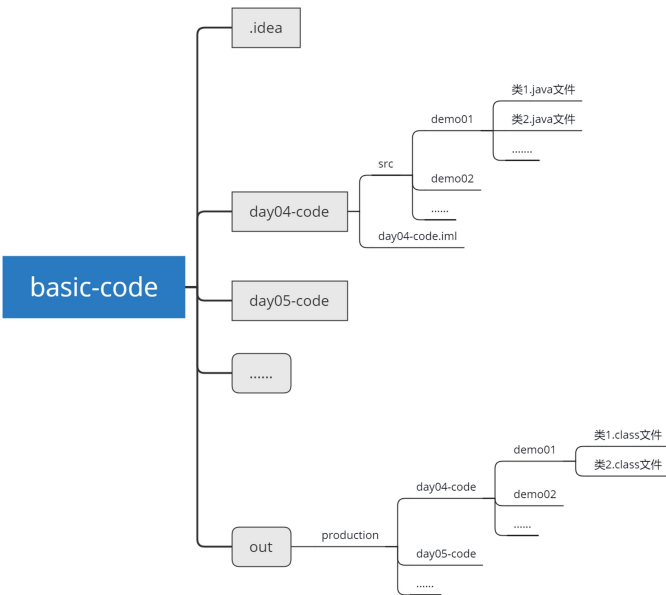
1.IntelliJ IDEA的项目结构



如上图所示，IDEA中的项目结构主要分为四层：

- (1).项目-project：黑马商城项目、Luxvision项目等等。（将整个基础班的代码作为一个项目，名为basic-code）
- (2).模块-Module：可以在项目里创建多个模块，将项目各个部分的功能交给不同的模块去完成。（在basic-code项目里面，将每一天的代码作为一个模块，如：day04-code、day05-code等）
- (3).包-package：可以在一个模块里创建多个包。所谓包，就是文件夹，用来对类文件进行管理。即使用包对源代码文件进行分门别类的管理。（在day04-code模块下创建包demo01、demo02等）
- (4).类-class：可以在包里创建多个类，一个类对应的生成一个.java源文件。（在包demo01、demo02里面创建多个不同的类1、类2等）

2.IntelliJ IDEA的文件目录



以基础班代码 basic-code 项目为例：

- (1).新建一个项目名为basic-code，则自动创建一个名为basic-code的文件夹，文件内部结构如图所示。
- (2)..idea文件夹：用来保存整个项目的IDEA配置文件。
- (3).在basic-code项目里面，新创建一个模块名为day04-code，就生成一个名为day04-code的文件夹。文件内部结构如图所示。
 - o day04-code.iml文件：用来保存该模块的IDEA配置信息
 - o src文件夹：在src里创建一个包就会生成一个文件夹；创建一个类，就生成一个.java源文件。所以在src里创建包和类，会生成相应的文件结构
- (4).out文件夹：用来保存basic-code项目下的所有.java文件编译生成的.class文件。其中的文件目录结构如图所示。

3.IntelliJ IDEA的常用快捷键

Alt+Enter	导入包、自动修正代码
Ctrl+Y	删除光标所在行
Ctrl+D	复制光标所在行内容，插入光标位置下面
Ctrl+Alt+L	格式化代码
Ctrl+/ Ctrl+Shift+/ Alt+Ins	单行注释，再按取消注释 多行注释，再按取消注释 自动生成代码(get、set等方法)
Alt+Shift+上下箭头	移动当前代码行

二.方法

1.方法的定义

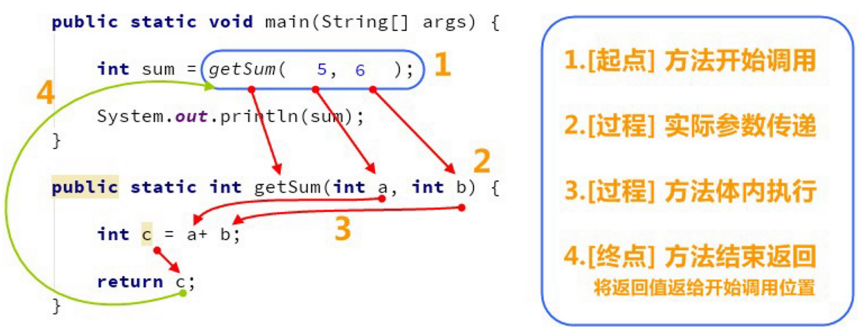
方法的定义格式如下：

```
修饰符 返回值类型 方法名称(参数类型 参数名称，...) {  
    方法体;  
    return 返回值;  
}
```

- 修饰符：目前固定写法 public static
- 返回值类型：方法最终返回数据的数据类型，由return后的返回值的类型来决定。若该方法没有返回值，则返回值类型为void。
- 方法名：要符合标识符的命名规则和命名规范，方法和变量的命名规范均是小驼峰式。
- 方法体的代码：该方法要实现的功能代码，可以包含任意条语句。
- 参数列表：定义相应的变量来接收进入方法的数据。参数如果有多个，使用逗号进行分隔。
- return语句：停止当前方法，遇到return则后面的代码均不执行，然后将后面的返回值返回给方法的调用处。
 - a.对于一个void没有返回值的方法：不能写return后面的返回值，只能写return自己用来结束方法，且此时return也可以省略不写。
 - b.对于一个有返回值的方法：方法结束处必须要有return语句来将返回值带出。

#.注意：方法里面的变量(包括形式参数变量)，其作用域仅限于该方法的内部。即一个方法里面的变量仅限于该方法内部使用，且一个类下面的各个方法里面的变量没有关系，可以重名。

2.方法的调用



一旦开始调用方法，程序立马跳转到方法的定义处。首先将调用语句的实际参数传递给形式参数接收，然后执行方法体的代码，最后使用return语句结束方法并将方法的返回值返回给方法的调用处。

3.方法的重载

在同一个类下定义多个方法时，允许它们的名称一样，但是形式参数列表不一样。调用有重载的方法时，具体调用重载中的哪一个方法，是看实际参数列表与哪一个形式参数列表相匹配。好处：只需要记住一个方法名称，就可以调用该方法实现多个类似的功能。

(1).构成方法重载的条件：方法名相同，形参列表不同即可构成重载。形参列表不同可分为以下几种情况：

- 参数个数不同可以构成重载
- 参数个数相同，但是参数类型不同可以构成重载

```
public static int sum(int a, int b) {  
    return a + b;  
}  
  
public static int sum(double a, int b) {  
    return a + b;  
}
```

(2).方法重载与下列因素无关：参数名称、方法的修饰符、返回值类型