

Notes from Well House Consultants

These notes are written by Well House Consultants and distributed under their Open Training Notes License. If a copy of this license is not supplied at the end of these notes, please visit

*<http://www.wellho.net/net/whcotnl.html>
for details.*

1.1 Well House Consultants

Well House Consultants provides niche training, primarily but not exclusively in Open Source programming languages. We offer public courses at our training centre and private courses at your offices. We also make some of our training notes available under our "Open Training Notes" license, such as we're doing in this document here.

1.2 Open Training Notes License

With an "Open Training Notes License", for which we make no charge, you're allowed to print, use and distribute these notes provided that you retain the complete and unaltered license agreement with them, including our copyright statement. This means that you can learn from the notes, and have others learn from them too.

You are NOT allowed to charge (directly or indirectly) for the copying or distribution of these notes, nor are you allowed to charge for presentations making any use of them.

1.3 Courses presented by the author

If you would like us to attend a course (Java, Perl, Python, PHP, Tcl/Tk, MySQL or Linux) presented by the author of these notes, please see our public course schedule at

<http://www.wellho.net/course/index.html>

If you have a group of 4 or more trainees who require the same course at the same time, it will cost you less to have us run a private course for you. Please visit our onsite training page at

<http://www.wellho.net/course/otc.html>

which will give you details and costing information

1.4 Contact Details

Well House Consultants may be found online at

<http://www.wellho.net>

graham@wellho.net

technical contact

lisa@wellho.net

administration contact

Our full postal address is

404 The Spa

Melksham

Wiltshire

UK SN12 6QL

Phone +44 (0) 1225 708225

Fax +44 (0) 1225 707126

Class Access

In order to provide the security of data and methods within classes, Java limits access through "the four Ps" – private, package, protected and public. Correct design of your objects and correct specification of the right "P" ensures that you can make all necessary methods and data available to users of your class, while at the same time hiding your internal workings.

| | |
|--|---|
| <i>Private, public, protected.</i> | 4 |
| <i>Inner classes</i> | 6 |
| <i>"finalize" method.</i> | 9 |

2.1 Private, public, protected

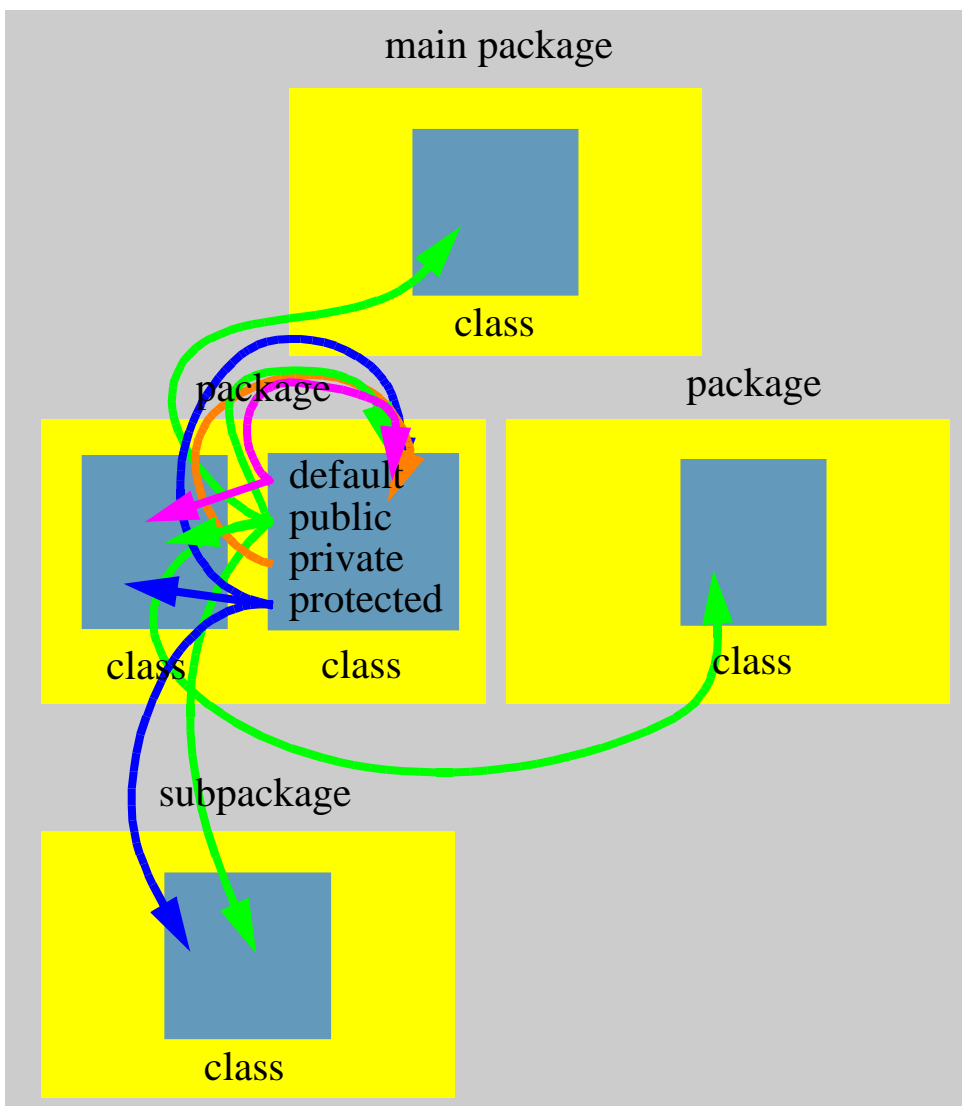
Recall that when we packaged up our *pack* class we had to declare the constructors as **public** so that they could be accessed from our main method in our default package? Class methods which are to be made available to everyone must be declared **public**.

Classes which are not declared as **public** (nor as **private** or **protected**) are available to any class in the same package.

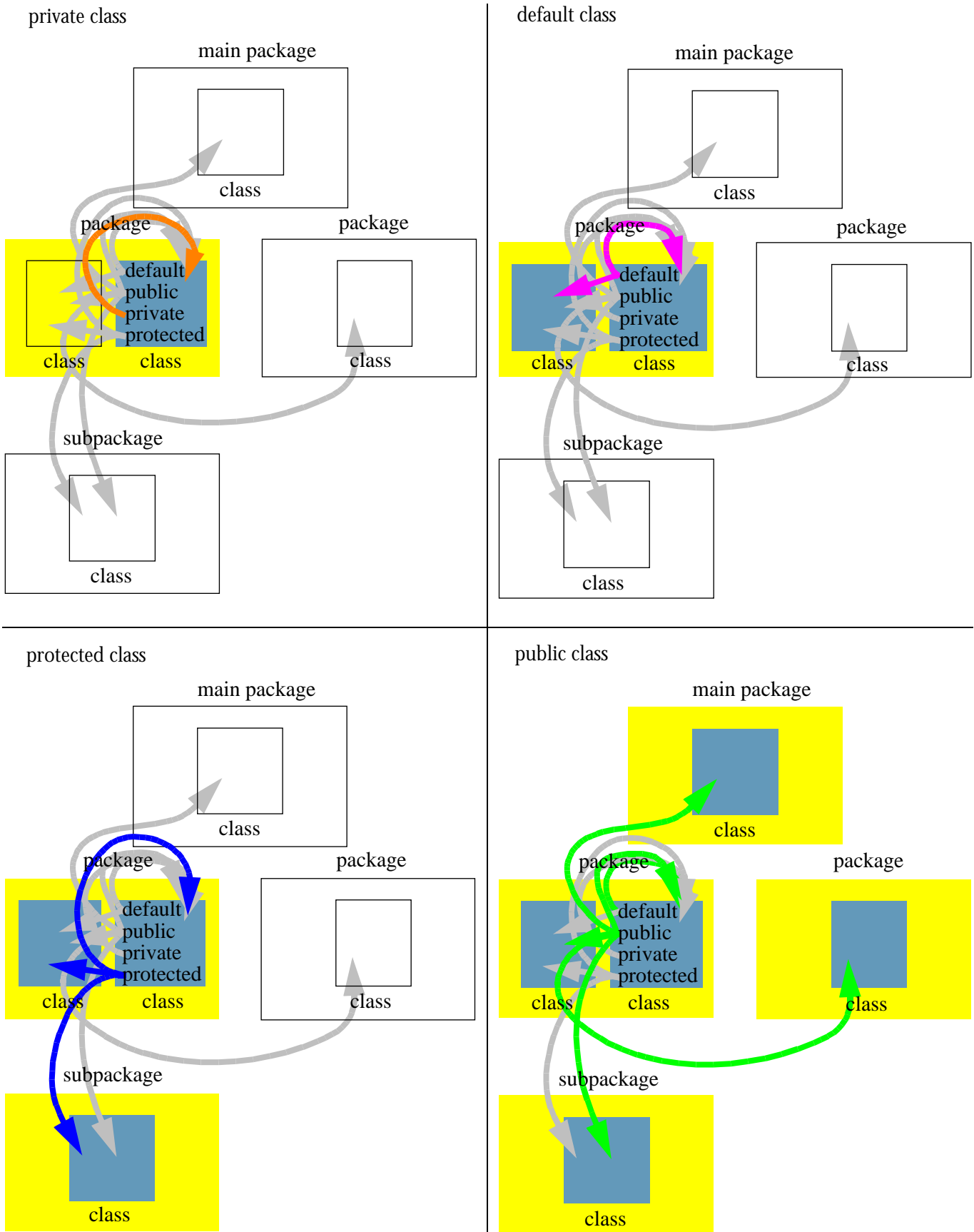
Private classes are only available within the class in which they are defined. They are very much internal methods that the author wants to make very certain cannot be used directly from outside.

Protected classes are available to any class in the same package (just like the default), plus any subclasses. They are not, however, available to any class as are **public** classes. In other words, they are a "halfway house" between the default and **public**.

It's quite simple really!



The diagram's a bit confusing though. Let's see ...



2.2 Inner classes

One class per source file can get a little long-winded and there might be times you want to define a class purely for use from within another class. This can be done with an **inner class**.¹

Modifying (once again) our *axford*/card pack example, we'll add a class for individual cards; a pack will now be made up from a number of card objects rather than a number of simple integers. We'll provide constructors and other methods for accessing cards, of course ...

```
// Well House Consultants2004.
/** inner classes */
public class luckington
{
    public static void main(String[] args)
    {

        System.out.println("Short pack, 24 cards removed");
        pack_5 my_pack = new pack_5(1,0,6);
        System.out.println(my_pack.ncards);
        my_pack.print();
    }
}

// Well House Consultants2004.
/** Card pack / Inner class card */
public class pack_5
{

// Instance Variables

    card [] cards;
    int ncards;

// Constructors

    pack_5(int ndecks, int njokers, int shorten)
    {
        ncards = (((13 - shorten) * 4 ) * ndecks) + njokers ;
        cards = new card [ncards];
        int n_at = 0;
        for (int k=0;k<njokers;k++)
            cards[n_at++]=new card(0);
        for (int i=0;i<ndecks;i++) {
            for (int k=shorten;k<13;k++) {
                for (int j=0;j<4;j++) {
                    cards[n_at++]=new card(j+4*k+1);
                }
            }
        }
    }
}

// method to access data
    public void print()
    {
        for (int k=0;k<ncards;k++)
```

¹ introduced at release 1.1 of Java

```
        {
        System.out.print(" " + cards[k].getcard());
        if ((k+1) % 10 == 0 || k == ncards-1)
            System.out.println();
        }
    }

// Inner class "card"
    class card
    {
        private int value;

// Constructor
        public card(int f_value)
        {
            value = f_value;
        }

// access data
        public int getcard()
        {
            return value;
        }
    }
}
```

In our example, the inner class is called *card* and we refer to the `pack_5` class as its "top-level" class.

You noticed that our definition of the inner class is made within the definition of the top-level class:

```
public class pack_5
{
```

[instance variables and methods for pack_5]

```
    class card
    {
```

[instance variables and methods for card]

```
    }
}
```

If you wish to access the methods of class *card* from outside class *pack_5* you could do so by:

- Ensuring that they were given enough access privileges (probably **public**).
- Referring to them only through an object of type *pack_5*.

```
pack_5 showyou = new pack_5(1,0,0);
pack_5.card played = showyou.new card(0);
```

To confirm the operation of our *luckington* / *pack_5* classes:

```
seal% java luckington
Short pack, 24 cards removed
28
 25 26 27 28 29 30 31 32 33 34
 35 36 37 38 39 40 41 42 43 44
 45 46 47 48 49 50 51 52
seal%
```

2.3 "finalize" method

Just as all instances that are going to be used are created, so all instances will be destroyed and their resources released at some time before your Java program exits.

Instance variables which go "out of scope" for example, will no longer be accessible and Java will release the memory for reuse. If you want specific actions taken when the memory is recovered, you may supply a **finalize** method.

Let's add a **finalize** method into our `pack_5` class:

```
// finalize method

protected void finalize()
{
    System.out.println("pack containing " +
        ncards + " cards released.");
}
```

WARNING: you should not rely on the **finalize** method to perform actions that are necessary as soon as an object is released. The Java Virtual Machine will only get rid of objects when it needs to reuse the memory that they are occupying.

Exercise

Add a method to your weather station class to return the temperature, as a string, to two decimal places.

Have the method call a static method to actually do the formatting of the number and ensure that this method can only be accessed from within the class.

Our example answer is cello

Our example answer is xylophone

Sample

```
seal% java xylophone
Stations defined:

Tiree
Ronaldsway
Channel light vessel automatic
Greenwich automatic
Total Weather Station Count: 4

Please give name of interestTiree
Matched Tiree
Temperature 31.50°C
seal% java xylophone
Stations defined:

Tiree
Ronaldsway
Channel light vessel automatic
Greenwich automatic
Total Weather Station Count: 4

Please give name of interestRonaldsway
Matched Ronaldsway
Temperature -17.60°C
seal%
```

For Advanced Students

Create an inner class to hold a wind speed object.

License

*These notes are distributed under the **Well House Consultants Open Training Notes License**. Basically, if you distribute it and use it for free, we'll let you have it for free. If you charge for its distribution of use, we'll charge.*

3.1 Open Training Notes License

Training notes distributed under the **Well House Consultants Open Training Notes License** (WHCOTNL) may be reproduced for any purpose PROVIDE THAT:

- This License statement is retained, unaltered (save for additions to the change log) and complete.
- No charge is made for the distribution, nor for the use or application thereof. This means that you can use them to run training sessions or as support material for those sessions, but you cannot then make a charge for those training sessions.
- Alterations to the content of the document are clearly marked as being such, and a log of amendments is added below this notice.
- These notes are provided "as is" with no warranty of fitness for purpose. Whilst every attempt has been made to ensure their accuracy, no liability can be accepted for any errors of the consequences thereof.

Copyright is retained by Well House Consultants Ltd, of 404, The Spa, Melksham, Wiltshire, UK, SN12 6QL - phone number +44 (1) 1225 708225. Email contact - Graham Ellis (graham@wellho.net).

Please send any amendments and corrections to these notes to the Copyright holder - under the spirit of the Open Distribution license, we will incorporate suitable changes into future releases for the use of the community.

If you are charged for this material, or for presentation of a course (Other than by Well House Consultants) using this material, please let us know. It is a violation of the license under which this notes are distributed for such a charge to be made, except by the Copyright Holder.

If you would like Well House Consultants to use this material to present a training course for your organisation, or if you wish to attend a public course is one is available, please contact us or see our web site - <http://www.wellho.net> - for further details.

Change log
Original Version, Well House Consultants, 2004

Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____

License Ends.