

Lab sheet:
Creating your own Classes
Setter: Katrin Lohan

Attempt Task 3A in the tutorial and finish Task 3B in the lab.

Task 3A: Create a class "Person"

(Also see exercises 2.66-2.73 in BlueJ Book/ 5th Ed.)

- 1) Create **new class** with Eclipse:
 - Download the lab 5 src code from vision, into your eclipse.
 - Create a new class under "SD1code/src/lab4" by right-clicking on the folder and select New... → Class. Call this class "**Person**".
 - Eclipse will automatically create an outer wrapping of a class.
- 2) Write out definitions for the following **fields**:
 - A field called "**name**" of type **String**
 - A field of type **int** called "**age**"
- 3) Write a **constructor** for the class Person:
 - The constructor should take two **parameters**: the first is of type **String** called "**myName**". The second is of type **int** and is called "**myAge**".
 - The body of the constructor should assign the value of the first parameter to the field **name**; the second should set the field **age**.
- 4) Add the corrected version of the following method (1 error!).

```
public void getAge()  
{  
    return age;  
}
```
- 5) Write an **accessor method** called **getName** that returns the value of the field **name**, whose type is **String**.
- 6) Write a **mutator method** called "**setAge**" that takes a single parameter of type **int** and sets the value of the field **age**.
- 7) Write a method called "**printDetails**" that prints out the value of the field **name**. It should print a single String in the form: *The name of this person is [name_value]*.
- 8) Write a **main method** to test your application. You can set the parameters to any value. This method creates a **new instance** of the class Person. We will cover how to create new instances in more detail in the next class. You can

run the class by right-clicking on the file and select “run as” → Java Application.

```
public static void main( String[] args ){
    Person p=new Person( "Eliza", 66);
    p.printDetails();
}
```

Task 8B: Create a class “Book”

(Also see exercises 2.83-2.85, 2.87, 2.89 in BlueJ Book.)

- 1) In your folder “SD1code/lab5” you should already have the outline of a class cladded `Book`. Open the class in Eclipse and double-check that:
 - The class already has two **fields**: `author` and `title`, both of type `String`.
 - The **constructor** takes two **parameters**, which initialize the two fields mentioned above.
- 2) Add two **accessor methods** to the class – `getAuthor` and `getTitle` – that return the author and title fields.
- 3) Add two **methods** `printAuthor` and `printTitle`, which print the respective fields to the terminal window, e.g. “The title of the book is: [title_value]”
- 4) Add a **field** called “`pages`” of type `int`.
 - It’s initial value should be set through the constructor (similar to the fields `author` and `title`).
 - Include an appropriate `getPages` accessor method.
- 5) Add a further field, `refNumber`, to the Book class. This field can store a reference number for a library, for example.
 - It should be of type `String` and initialized to the zero length string in the constructor. Note that it’s initial value is not passed as a parameter!
 - Instead, define the a **mutator method** with the following header:

```
public void setRefNumber(String ref)
```

The body of this method should assign the value of the parameter to the `refNumber` field.
 - Also add a corresponding `getRefNumber` **accessor method**.
- 6) Modify your `setNumber` mutator method so that it sets the `refNumber` field only if the parameter is a string of at least 3 characters. If it’s less than three, then print an **error message** and leave the field unchanged.
- 7) *Challenge exercise*: write a **main method** to test your application.