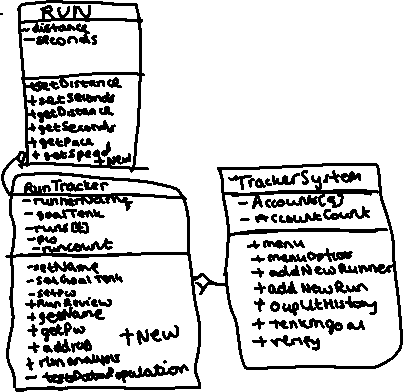
**Skeleton Program**

You will be receiving a skeleton program to research over the holidays. You should use the following pages to make notes on the program. Identify what it does, how it works, any errors and what improvements you would make.

**Explain the system, its purpose, what it does, data input, data returned**

|  |
| --- |
| The purpose of this system is to help Adidas sports system. It allows the user to input their name, add a run they did in metres and seconds, outputs its history and provides a 10km goal analysis. Data input: name, password, distance, speed. Data returned: 10km analysis, and returns the run history when asked, average speed, average pace. |

**Class Diagram**



|  |
| --- |
|  |

**Explain any inheritance**

|  |
| --- |
| There is no inheritance, just association aggregation and static. Association aggregation means that if RUN is deleted then RUN TRACKER and TRACKER SYSTEM are not deleted. Static method (Convert) can be run without being instantiated. |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| Run | Properties/ Variables: distance, seconds, newValue  Methods: SetDistance(s), SetSeconds(s), GetDistance(f), GetSeconds(f), GetPace(f), GetSpeed(f), OutputRun(s)  Distance – this stores the distance they ran (entered in metres)  Seconds – this stores the time it took for them to run it in (entered in seconds)  Get Distance – [ByVal newValue] literally gets the distance  Get Seconds – [ByVal newValue] literally gets the timing  Set Distance – saves the distance  Set Seconds – saves the timing  Get Pace – (seconds/60) means in minutes (distance/1000) means in kilometres   * Time / Distance   Get Speed - |
|  |  |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| TrackerSystem |  |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| RunTracker |  |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Class**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |
| --- | --- |
| Identifier | What it does (inheritance, aggregation etc) |
| Converter |  |

**Properties**

List the variables involved in the program, their role, any validation performed or required, scope, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Role | Validation | Scope (G/L) | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Methods**

List the subroutines involved in the program, parameters, return type, type, and what they do

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier | Type  (F/P) | Parameters | Return | What it does |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Errors**

|  |  |  |
| --- | --- | --- |
| **Where** | **Why** | **Solution** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Improvements**

|  |  |  |
| --- | --- | --- |
| **Where** | **Why** | **Solution** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |