Banana Technologies Code conventions

1. Naming

1.1 Use meaningful names.

Use descriptive names for all identifiers (names of classes, variables and methods. Avoid ambiguity. Avoid abbreviations.
Simple mutator methods should be named set*Something*(…).
Simple accessor methods should be named get*Something*(…).
Accessor methods with Boolean return values are often called is*Something*(…), for example isEmpty().

1.2 Class names start with a capital letter.

1.3 Class names are singular nouns.

1.4 Methods and variable names start with lowercase letters.

All the – class, method and variable names – use capital letters in the middle to increase readability of compound identifiers, e.g. numberOfItems.

1.5 Constants are written in UPPERCASE.

Constants occasionally use underscores to indicate compound identifiers:
MAXIMUM_SIZE

2. Layout

2.1 One level of indentation is four spaces.

2.2 All statements within a block are intended one level.

2.3 Braces for classes and methods are alone on one line.

The braces for class and method blocks are on separate lines and are at the same indentation level, for example:

```
public int getAge()
{
    statements
}
```

2.4 Also for all other blocks, are alone on one line.

The braces for all other blocks are on separate lines and are at the same indentation level, for example:

```
while(condition)
```

```
{
    statements
}


if(condition)
{
    statements
}
else
{
    statements
}
```

2.5 Inline statements.

Usage of an inline statement is allowed without braces if the body only contains one line of code

Example:

```
if(condition) statement;
```

2.6 Use a space before the opening brace of a control structure's block.

2.7 Use a space around operators.

Example:

incorrect: `$var=5+5;`

correct: `$var = 5 + 5;`

2.8 Use a blank line between methods (and constructors).

Use blank lines to separate logical blocks of code. This means at least between methods, but also between logical parts within a method.