Module 3 Lab 3A

Name: Clayton Black
Date: 09-26-2019
Assignment Name: Module 3 Lab 3A
Assignment Brief: inheritance
Sources:

- https://ccse.kennesaw.edu/fye/pseudocode/pseudocodeguide.php

Main / client / Tester

```
CLASS Main
BEGIN

    CREATE randomWords[10] ## This will be a static array of words choosen at random.
    CREATE taxRates ← { .05, .06, .07, .08, .09, .1}

    METHOD Main
    BEGIN
        CREATE stores[3]

        CREATE stores[0] as Store
        CREATE stores[1] as YarnStore
        CREATE stores[1] as YarnStore

        FOREACH store in stores
            PRINT "Pre Changes: " + store

            CREATE newName ← randomWords[random number] + " " + randomWords[random number]
            store.setName(newName)

            store.setTaxRate(taxRates[random number])

            PRINT "Store Name: " + store.getName()
            PRINT "Tax Rate: " + store.getTaxRate()
```

```
            IF store is a YarnStore THEN
                store.setAvgPricePerSkein(random int)
                store.setSkeinsSoldPerYear(random int)

                PRINT "Skeins Sold Per Year: " + store.getSkeinsSoldPerYear()
                PRINT "Average Price Per Skein: " + store.getAvgPricePerSkein()
                PRINT "Average Taxes Per Year: " + store.avgTaxesPerYear()
            ENDIF

        CREATE s ← random number between 0 and length of stores
        PRINT "This store is the same as store[" + s + "]: " + store.equals(stores[s])

        PRINT "Post Changes: " + store

    END METHOD

END CLASS
```

Store Class

```
CLASS Store
BEGIN
    CREATE name
    CREATE taxRate

    CONSTRUCTOR Store(parameter: name, taxRate)
    BEGIN
        this.name ← name
        this.taxRate ← taxRate
    END CONSTRUCTOR

    METHOD getName()
    BEGIN
        return name
    END METHOD

    METHOD getTaxRate()
    BEGIN
        return taxRate
```

```
    END METHOD

    METHOD setName(parameter: name)
    BEGIN
        this.name ← name
    END METHOD

    METHOD setTaxRate(parameter: taxRate)
    BEGIN
        this.taxRate ← taxRate
    END METHOD

    METHOD equals(parameter: o)
    BEGIN
        IF this == o THEN RETURN TRUE ENDIF
        IF 0 == null OR o and this are different types of objects THEN RETURN FALSE

        IF this.taxRate == o.taxRate && this.name == o.name THEN RETURN TRUE
    END METHOD

    METHOD toString()
    BEGIN
        RETURN "Store{name=" + name + ", taxRate=" + taxRate + "}"
    END METHOD

END CLASS
```

YarnStore Class

```
CLASS YarnStore
BEGIN
    CREATE skeinsSoldPerYear
    CREATE avgPricePerSkein

    CONSTRUCTOR YardStore(parameter: name, taxRate, skeinsSoldPerYear, avgPricePerSkein)
    BEGIN
        super(name, taxRate)
        this.skeinsSoldPerYear ← skeinsSoldPerYear
        this.avgPricePerSkein ← avgPricePerSkein
    END CONSTRUCTOR
```

```
    METHOD getSkeinsSoldPerYear()
    BEGIN
        return skeinsSoldPerYear
    END METHOD

    METHOD getAvgPricePerSkein()
    BEGIN
        return avgPricePerSkein
    END METHOD

    METHOD setSkeinsSoldPerYear(parameter: skeinsSoldPerYear)
    BEGIN
        this.skeinsSoldPerYear ← skeinsSoldPerYear
    END METHOD

    METHOD setAvgPricePerSkein(parameter: avgPricePerSkein)
    BEGIN
        this.avgPricePerSkein ← avgPricePerSkein
    END METHOD

    METHOD equals(parameter: o)
    BEGIN
        IF this == o THEN RETURN TRUE
        IF o == null || this and o are of differnet types THEN RETURN FALSE
        IF  this.skeinsSoldPerYear == o.skeinsSoldPerYear && this.avgPricePerSkein == o.avgPricePerSkein THEN RETURN TRUE
    END METHOD


    METHOD toString()
    BEGIN
        RETURN "YarnStore{name=" + getName() + ", taxRate=" +
                getTaxRate() + ", skeinsSoldPerYear=" + skeinsSoldPerYear +
                ", avgPricePerSkein=" + avgPricePerSkein + "}"
    END METHOD

    METHOD avgTaxesPerYear(){
    BEGIN
        return (avgPricePerSkein * skeinsSoldPerYear) * getTaxRate()
    END METHOD

END CLASS
```