

```

MAIN
BEGIN
    PRINT "Please enter All-Inclusive Vacation's budget: "
    CREATE budget ← READ
    PRINT "Please enter All-Inclusive Vacation's destination: "
    CREATE destination ← READ
    PRINT "Please enter All-Inclusive Vacation's brand: "
    CREATE brand ← READ
    PRINT "Please enter All-Inclusive Vacation's rating: "
    CREATE rating ← READ
    PRINT "Please enter All-Inclusive Vacation's price: "
    CREATE price ← READ
    CREATE allInclusive ← NEW Vacation.All_Inclusive(budget, destination,
brand, rating, price)
    PRINTLINE
    PRINTLINE allInclusive.ToString()
    PRINTLINE
    PRINTLINE allInclusive.sendBudget()
    PRINTLINE
    PRINT "Please enter PiecemealVacation Vacation's budget: "
    budget ← READ
    PRINT "Please enter PiecemealVacation's destination: "
    destination ← READ
    CREATE expenses ← NEW List<string>()
    CREATE costs ← NEW List<int>()
    WHILE TRUE
        PRINT "Please enter PiecemealVacation's items: "
        expenses.Add(READ)
        PRINT "Please enter Hotel cost: "
        costs.Add(READ)
        PRINT "Have you finished entering all items? "
        IF (READ == "Yes") THEN
            BREAK
        ENDIF
    ENDWHILE
    CREATE piecemeal ← NEW Vacation.Piecemeal(budget, destination,
expenses.ToArray(), costs.ToArray())
    PRINTLINE
    PRINTLINE piecemeal.ToString()
    PRINTLINE
    PRINTLINE piecemeal.sendBudget()
END MAIN

```

```

CLASS Vacation
BEGIN
    budget ← 0
    destination ← ""

    METHOD ToString()
    BEGIN
        RETURN this.GetType().Name + ":\n Budget: " + budget + "\n
Destination: " + destination
    END METHOD

```

```

METHOD Budget (parameter: cost)
BEGIN
    IF (budget >= cost) THEN
        RETURN "This vacation was $" + (budget - cost) + " under
budget."
    END IF
    RETURN "This vacation was $" + (cost - budget) + " over budget."
END METHOD

CLASS All_Inclusive : Vacation
    brand ← ""
    rating ← 0
    price ← 0

    CONSTRUCTOR All_Inclusive (parameter: newBudget, newDestination,
newBrand, newRating, newPrice)
        budget ← newBudget
        destination ← newDestination
        brand ← newBrand
        rating ← newRating
        price ← newPrice
    END CONSTRUCTOR

    METHOD ToString()
    BEGIN
        RETURN "The budget is $" + budget + ", the destination is " +
destination + ", the brand is " + brand + ", the rating is " + rating + "
stars, and the price is $" + price + "."
    END METHOD

    METHOD sendBudget() => RETURN Budget(price)

END CLASS

CLASS Piecemeal : Vacation
    expenses ← NULL
    costs ← NULL

    CONSTRUCTOR Piecemeal (parameter: newBudget, newDestination,
newExpenses, newCosts)
        budget ← newBudget
        destination ← newDestination
        expenses ← newExpenses
        costs ← newCosts
    END CONSTRUCTOR

    METHOD ToString()
        CREATE temp ← "The budget is $" + budget + ", the destination
is " + destination + ", and the items needed for the vacation are:"
        FOR i ← 0 to expenses.Length
            IF (i == expenses.Length - 1) THEN
                temp ← temp + " and"
            ENDIF

```

```

        temp ← temp + " a " + expenses[i] + " that costs $" +
costs[i]
        IF (i == expenses.Length - 1) THEN
            temp ← temp + "."
        ELSE
            temp ← temp + ","
        ENDIF
    ENDFOR
    RETURN temp
END METHOD

METHOD sendBudget()
    CREATE sum ← 0
    FOREACH x in costs
        sum ← sum + x
    END FOREACH
    RETURN Budget(sum)
END METHOD
END CLASS
END CLASS

```