CookieEvent class – The purpose of this class is to calculate the total number of sales made and the total number of cookies sold. It will allow a string representation of the event name, event type, maximum number of cookies allowed to be sold, the number of cookies sold, and the number of sales in the cookie event.

|  |  |
| --- | --- |
| (-) eventName: String | The name of the cookie selling event. |
| (-) eventType: String | The cookie selling event type, either for profit and not for profit. |
| (-) maxCookiesSold : int | The maximum number of cookies allowed to be sold (Default: 206). |
| (-)cookiesSold: int | The number of cookies sold (will initially be 0). |
| (-)saleCounter: int | The number of sales in the cookie selling event (will initially be 0). |
|  |  |
|  |  |
| (+)CookieEvent(eventName: String, eventType: boolean) | Constructs a CookieEvent object that will chain the default maxCookiesSold and the specified eventName, and eventType to another CookieEvent object. |
| (+)CookieEvent(eventName: String, eventType: boolean, cookiesSold: int, saleCounter: int, maxCookiesSold: int) | Constructs a CookieEvent with a specified eventName, eventType, cookiesSold, saleCounter, and maxCookiesSold. |
| (+)getEventName(): String | Returns the eventName of this CookieEvent. |
| (+)getEventType(): Boolean | Returns the eventType of this CookieEvent. |
| (+)getMaxCookiesSold(): int | Returns the maxCookiesSold of this CookieEvent. |
| (+)getCookiesSold(): int | Returns the cookiesSold of this CookieEvent. |
| (+)getSaleCounter(): int | Returns the saleCounter of this CookieEvent. |
| (+)setEventName(eventName: String): boolean | Returns true if eventName is not null and sets a eventName for this CookieEvent or returns false. |
| (+)setEventType(eventType: String): Boolean | Returns true if eventType is “for profit” or “not for profit” and sets a eventType for this CookieEvent or returns false. |
| (+)setMaxCookiesSold(maxCookiesSold: int): Boolean | Returns true if maxCookiesSold is between 1 and 923 inclusive and sets a maxCookiesSold for this CookieEvent or returns false. |
| (+)setSaleCounter(saleCounter: int): void | Sets a saleCounter for this CookieEvent. |
| (+)setCookiesSold(cookiesSold: int): void | Sets a cookiesSold for this CookieEvent. |
| (+)doubleUp(singleSale: int): int | Doubles the cookie value if the event type is not for profit. returns an int. |
| (+)cookieSale(singleSale: int): Boolean | Adds the number of cookies that want to be bought (singleSale) to the number of cookies already sold (cookiesSold) if there are enough cookies to sell. Returns true if there are enough to sell. |
| (+)countTheSales(): void | Counts the number of sales using saleCounter. Return is void. |
| (+)toString(): String | Returns a string representation of the cookie selling event, including event name, event type, maximum number of cookies allowed to be sold, the number of cookies actually sold, and the number of sales made. |

**Method:** getEventName

**Purpose:** The purpose is to return the eventName attribute.

**Inputs:** none.

**Return:** eventName: String – The event name for the cookie event.

**Method:** getEventType

**Purpose:** The purpose is to return the eventType attribute.

**Inputs:** none.

**Return:** eventType: String – The event type for the cookie event, either for profit or not for profit.

**Method:** getMaxCookiesSold

**Purpose:** The purpose is to return the maxCookiesSold attribute.

**Inputs:** none.

**Return:** maxCookiesSold: int – The maximum number of cookies allowed to be sold, default is 206.

**Method:** getCookiesSold

**Purpose:** The purpose is to return the cookiesSold attribute.

**Inputs:** none.

**Return:** cookiesSold: int – The number of cookies sold at the cookie event.

**Method:** getSaleCounter

**Purpose:** The purpose is to return the saleCounter attribute.

**Inputs:** none.

**Return:** saleCounter: int – The number of sales at the cookie event.

**Method:** setEventName

**Purpose:** Sets the event name if it is not null or not blank and returns Boolean value based on the validation.

**Inputs:** eventName: String – The name of the cookie event.

**Return:** Boolean – If eventName passes the validation, the Boolean value will be true.

**Method:** setEventType

**Purpose:** Sets the event type if it’s equal to “for profit” or “not for profit” and returns a Boolean value based on the validation.

**Inputs:** eventType: String – The type of cookie event.

**Return:** Boolean – if eventType passes the validation, the Boolean value will be true.

**Method:** setMaxCookiesSold

**Purpose:** Sets the maximum cookies sold if it is between 1 and 923 inclusive and returns a Boolean value based on the validation.

**Inputs:** maxCookiesSold: int – The maximum number of cookies allowed to be sold at the cookie event.

**Return:** Boolean – if maxCookiesSold passes the validation, the Boolean value will be true.

**Method:** setCookiesSold

**Purpose:** Sets the total cookies sold at the event if it is greater than 0 and returns a Boolean value based on the validation.

**Inputs:** cookiesSold: int – The total number of cookies sold at the cookie event.

**Return:** Boolean – if cookiesSold passes the validation, the Boolean value will be true.

**Method:** doubleup

**Purpose:** Doubles the value of singleSale if eventType equals “not for profit”.

**Inputs:** singleSale: int – A passed integer value that is the number of cookies that want to be bought at a single sale.

**Return:** int – an integer that will represent how many cookies will be sold at this single sale.

**Method:** cookieSale

**Purpose:** Adds the number of cookies that want to be bought (singleSale) to the number of cookies already sold (cookiesSold) if there are enough cookies to sell. This method will compare (maxCookiesSold – cookiesSold) to the value that is returned from the method doubleUp. If it passes that validation, then it will add the number of cookies to be bought to the cookies already bought.

**Inputs:** singleSale: int – A passed integer value that is the number of cookies that want to be bought at a single sale.

**Return:** Boolean – The method will return true if there are enough cookies to be sold and that the cookies that wanted to be sold are added to the sale number.

**Method:** countTheSales

**Purpose:** Counts the number of sales in the cookie event.

**Inputs:** none.

**Return:** void

**Method:** toString

**Purpose:** The method will return a string representation of the cookie selling event, including event name, event type, maximum number of cookies allowed to be sold, the number of cookies actually sold, and the number of sales made.

**Inputs:** none

**Return:** String – a formatted string with eventName, eventType, maxCookiesSold, CookiesSold, and singleSale.