```
// SyntaxItem.m

#import "SyntaxItem.h"

@implementation SyntaxItem

- free
{
    [regularExpression free];
    [matchedSampleText free];
    free(regularExpressionText);
    return [super free];
}

- read:(NXTypedStream *)stream
{
    [super read:stream];
    [regularExpression free];
    regularExpression = NXReadObject(stream);
    [matchedSampleText free];
    matchedSampleText = NXReadObject(stream);
    [matchedSampleText setTextColor:NXReadColor(stream)];
    return self;
}

- write:(NXTypedStream *)stream
{
    [super write:stream];
    NXWriteRootObject(stream, regularExpression);
    NXWriteRootObject(stream, matchedSampleText);
    NXWriteColor(stream, [matchedSampleText textColor]);
    return self;
}

- getRegularExpression
{
    return regularExpression;
}

- (char *)getRegularExpressionText
{
    int length = [regularExpression textLength];
    free(regularExpressionText);
    if (!(regularExpressionText = malloc(length + 1))) {
        NXRunAlertPanel("Error", "malloc() failed", 0, 0, 0);
        return "";
    }
    if ([regularExpression getSubstring:regularExpressionText start:0 length:length] <= 0) {
//      NXRunAlertPanel("Error", "Can't get regular expression substring", 0, 0, 0);
        return "";
    }
    regularExpressionText[length] = 0;
    return regularExpressionText;
}

- getMatchedSampleText
{
    return matchedSampleText;
}

- (void)setRegularExpression:re
{
    NXStream *stream = [re stream];
    [regularExpression free];
    NXSeek(stream, 0L, NX_FROMSTART);
    regularExpression = [[[Text alloc] init] readText:stream];
    [regularExpression setFont:[re font]];
}

- (void)setMatchedSampleText:text
{
    NXStream *stream = [text stream];
    [matchedSampleText free];
    NXSeek(stream, 0L, NX_FROMSTART);
    matchedSampleText = [[Text alloc] init];
    [matchedSampleText readText:stream];
    [matchedSampleText setFont:[text font]];
    [matchedSampleText setTextColor:[text textColor]];
}

@end
```