

## Zadanie 1 – iOS

### Wstęp:

Jeśli znasz objective-c możesz przejść do rozwiązywania zadania właściwego.

- przeczytaj jakikolwiek tutorial wprowadzający w składnię i sposób programowania w objective-c.
  - <http://blog.teamtreehouse.com/an-introduction-to-objective-c>
  - <http://rypress.com/tutorials/objective-c/introduction.html>
- przeczytaj "Programming with Objective-C". Rozdziały "About Objective-C" oraz "Defining Classes".  
<http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/ProgrammingWithObjectiveC.pdf>
- Instalacja narzędzi pomocnych w walidacji składni języka i kompilacji:

Użytkownicy	
Mac OSX	Windows oraz pozostałych systemów
Pobierz i zainstaluj z App Store'a <b>Xcode</b> <a href="https://itunes.apple.com/pl/app/xcode/id497799835?mt=12">https://itunes.apple.com/pl/app/xcode/id497799835?mt=12</a>	Pobierz i zainstaluj <b>GNUStep</b> <a href="http://gnustep.org/resources/sources.html">http://gnustep.org/resources/sources.html</a>  Lub skorzystaj z kompilatora online <a href="http://www.compileonline.com/compile_objective-c_online.php">http://www.compileonline.com/compile_objective-c_online.php</a>

### GNUStep konfiguracja w systemie Windows

Przewodnik opisujący szczegółowo konfigurację środowiska oraz kompilację plików.

<http://scottwhite.blogspot.com/2010/11/getting-started-programming-objective-c.html>

Ogólne kroki to:

1. instalacja następujących pakietów:

- gnustep-msys-system
- gnustep-core
- gnustep-devel

2. stworzenie pliku "**GNUmakefile**" ze skryptem budującym

- skopiowanie skryptu z powyższego przewodnika i zapisanie w katalogu projektu
- modyfikacja zmiennej **GNUSTEP\_MAKEFILES** w skrypcie i ustawienie odpowiedniej ścieżki do plików
- dodanie plików do zmiennej **TestApp\_OBJC\_FILES**

## Zadanie:

Stwórz klasę **PhoneNumber** z następującymi własnościami:

- inicjalizowana poprzez numer w postaci tekstu (użyć klasy **NSString**),
- posiada pole tylko do odczytu zwracające numer w postaci liczby,
- posiada pole tylko do odczytu zwracające numer kierunkowy (bez +),
- posiada pole tylko do odczytu zwracające sformatowany test w postaci "+48 123 456 789" (od prawej co trzeci znak spacja)
- metodę pozwalającą ustawić numer telefonu, powinna ona aktualizować wszystkie pozostałe pola.
- funkcję description zwracającą **NSString** z wartościami wszystkich pól klasy.

W nieomówionych kwestiach pozostawiamy uczestnikom pełną dowolność.

## Porady:

Do formatowania tekstu możesz użyć metody `stringWithFormat` z klasy `NSString`. Podczas formatowania tekstu możesz używać `%@` do wypisywania zawartości stringów i obiektów.

Więcej na ten temat:

<https://developer.apple.com/library/ios/documentation/cocoa/conceptual/Strings/Articles/FormatStrings.html>

### Przykładowy kod testowy

```
#import <Foundation/Foundation.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    PhoneNumber *pn = [[[PhoneNumber alloc] initWithNumber:@"715111444"]
    autorelease];
    NSLog(@"PhoneNumber: %@", pn);
    [pn setNumber:@"+48123456789"];
    NSLog(@"PhoneNumber: %@", pn);
    [pn setNumber:@"18456123789"];
    NSLog(@"PhoneNumber: %@", pn);
    [pool drain];
    return 0;
}
```