

# LABORATORIUM SYSTEMÓW MOBILNYCH

---

## STWORZENIE APLIKACJI WIZUALIZUJĄCEJ WEKTOROWE DANE PRZESTRZENNE

### I. Temat ćwiczenia

Stworzenie aplikacji na **Pocket PC**, która wykorzystywałaby stworzony w poprzednim ćwiczeniu moduł **MapPrepare** i umożliwiałaby wyświetlanie przykładowej mapy, jej płynne przewijanie i powiększanie.

### II. Wymagania

Wykonany na poprzednich zajęciach moduł **MapPrepare**.  
Znajomość podstaw tworzenia aplikacji mobilnych.

### III. Ćwiczenie

#### 1. Stworzenie nowego projektu Pocket PC

Należy stworzyć przykładowy projekt na **Pocket PC 2003**, zawierający 3 zakładki i na każdej z nich komponent typu **PictureBox** i przyciski przewijania i powiększania mapy.

Aplikacja powinna umożliwiać wczytanie mapy z pliku **MIF** lub **SHP** i konwersji danych do postaci relacyjnej – należy wykorzystać stworzony na jednym z poprzednich zajęć moduł do konwersji danych **MIF/SHP** do postaci relacyjnej.

Poszczególne (3) zakładki będą umożliwiały porównanie różnych sposobów wyświetlania mapy i testowanie interaktywności z użytkownikiem.

#### Pierwsza zakładka (brak optymalizacji):

- Przerobić moduł **MapPrepare** tak, aby za każdym razem przy przesunięciu lub powiększeniu mapy generował na nowo cały obszar wyświetlania (bez buforowania obszarów sąsiednich, generowany obszar ma mieć dokładnie taką wielkość jak obszar wyświetlania).

#### Druga zakładka (częściowo zoptymalizowane):

- Przerobić moduł **MapPrepare** tak, aby generował z każdej strony 2 razy większy obszar niż obszar wyświetlania, a dopiero przy dojściu do granic obszaru następowało ponowne generowanie obszaru.

#### Trzecia zakładka (pełne buforowanie z obszarami sąsiednimi):

- Wykorzystać ostateczną postać modułu **MapPrepare**, tak aby wykorzystać pełne możliwości optymalizacji.

W każdym z przypadków należy zauważyć różnice w interakcyjności pracy z aplikacją i szybkością generowania mapy.

## 2. Optymalizacja wyświetlania (płynność przełączania zdjęć)

Projektując aplikację Windowsową, w celu zapewnienia płynności przełączania zdjęć stosuje się tak zwany „double buffering”.

Niestety z uwagi na ograniczenia, aplikacje **Pocket PC** nie obsługują tej funkcji. Sprawia to, iż wyświetlanie lub podmienianie obrazu powoduje nieprzyjemny efekt migotania.

Aby temu zapobiec należy wykorzystać następujący „trick”:

- Stworzyć klasę **PictureBox2**, która dziedziczy z klasy **PictureBox**

```
public class PictureBox2 : PictureBox
{
    ...
}
```

- Stworzyć pustą metodę **OnPaintBackground** (aby zapobiec migotaniu tła)

```
protected override void OnPaintBackground(PaintEventArgs e)
{
}
```

- Stworzyć metodę **OnPaint**, która będzie wyświetlała aktualną (stworzoną wcześniej – wygenerowaną – bitmapę)

```
protected override void OnPaint(PaintEventArgs e)
{
    Graphics gr = e.Graphics;
    gr.DrawImage(img, this.ClientRectangle, srcRect, GraphicsUnit.Pixel);
}
```

**srcRect** jest obiektem typu **Rectangle**, który przechowuje aktualny prostokąt (**obszaru „5”**), który jest wyświetlany.

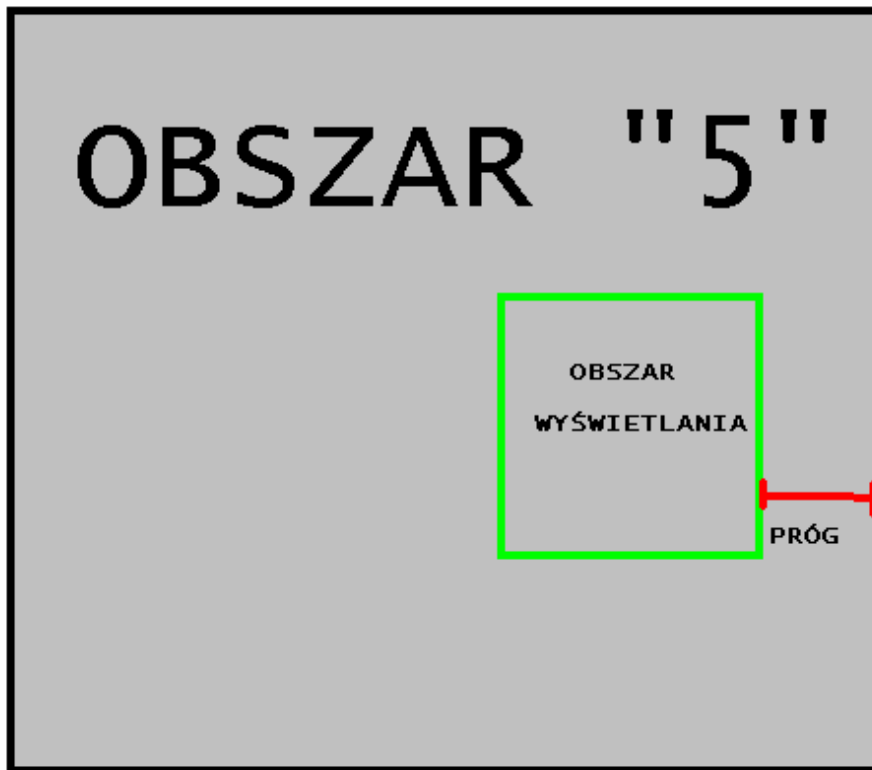
Jest to bardzo przydatne, gdyż pozwala wyświetlić tylko część bitmapy, która przechowuje obszar „5”, a tym samym płynne przewijanie mapy.

Należy pamiętać, aby po wykonaniu zmiany bitmapy przechowującej **obszar „5”** (pobranie za pomocą funkcji **MapPrepare.GetMainRegion**) i uaktualnieniu obszaru wskazywanego przez **srcRect** wywołać funkcję **Invalidate()** na danym **pictureBox**-ie i metodę **Refresh**.

```
this.pictureBox1.Invalidate();
this.pictureBox1.Refresh();
```

Dopiero to spowoduje przerysowanie i odświeżenie mapy.

Dla celów optymalizacji (przy zakładce 3-ciej) można założyć pewien próg (określający odległość granic obszaru wyświetlania od granic większego obszaru „5” przechowywanego jako bitmap-a) i dopiero gdy odległość spadnie poniżej konkretnej wartości progu wywoływać ponownie funkcję **MapPrepare.GetMainRegion**, która stworzy bardziej aktualny obszar (metodą „sklejania”) i pobierze w tle lub wygeneruje (wątki) nowe sąsiednie obszary.



Spowoduje to dodatkową, znaczną optymalizację płynności wyświetlania mapy.