

Politechnika Śląska  
Wydział Informatyki, Elektroniki i Informatyki

# Podstawy Programowania Komputerów

MiniCity - Symulator Miasta

---

autor	Jarosław Makówka
prowadzący	dr inż. Maciej Długosz
rok akademicki	2023/2024
kierunek	Informatyka
rodzaj studiów	SSI
semestr	2
termin laboratorium	czwartek, 13:30 – 15:00
sekcja	24
termin oddania sprawozdania	2024-07-01

---



## 1 Treść zadania

Proszę napisać program będący symulatorem miasta. Zadaniem gracza jest wytyczanie dróg, wyznaczanie stref mieszkalnych oraz stref do budowy zakładów pracy, a także specjalnych budynków komunalnych (np. szkół). Różne strefy posiadają różne parametry (np. różna wartość ziemi w strefach mieszkalnych (typu liczba rzeczywista) albo liczba osób mogących zostać zatrudnionych w danej strefie (liczba naturalna)). Program powinien implementować prosty budżet (obejmujący np. podatki, koszty budowy i utrzymania infrastruktury) oraz kilka (wybieranych przez użytkownika albo losowo) algorytmów wyznaczających zainteresowanie mieszkańców zamieszkaniem w mieście.

## 2 Uwagi techniczne

Programy powinny zostać napisane zgodnie z zasadami programowania obiektowego i wykorzystywać każdą z poniższych technik:

- klasy wraz z dziedziczeniem,
- plik wyjściowy z liczbami posortowanymi
- szablony (wzorce),
- co najmniej jeden wzorzec projektowy.

Dodatkowo należy wykorzystać co najmniej jedną z poniższych technik:

- wyjątki,
- dziedziczenie wielobazowe,
- przeciążanie operatorów (sensowne oraz więcej niż przeciążenie operatorów przypisania (`=`), porównania (`==`)).

## 3 Analiza zadania

Stworzyć symulator miasta, który pozwoli rozbudowywać miasto przez kolejne lata. Użytkownik będzie zarządzał budową infrastruktury w mieście. Co roku jest wyliczana ilość nowych mieszkańców.

### 3.1 Struktury danych

Użyto wektora wektorów, żeby przechowywać siatkę miasta.

### 3.2 Algorytmy

Algorytym wznaczenia zainteresowanie mieszkańców uważają biblioteki `random`.

### 3.3 Wzorzec projektowy

Użyto wzorca projektowego Strategia aby wybrać algorytm wuznaczący zainteresowanie mieszkańców życiem w mieście.

### 3.4 Szablony

Dwukrotnie użyto szablonów:

- Metoda `switch_zone` klasy `View` wywołuje metodę `build_zone` z odpowiednim parametrem typu: `Residential_zone` lub `Industrial_zone`.
- Szablon klasy `Zone` pozwala strefie mieszkalnej przechowywać wartość charakterystyczną w typie **int**, a strefie przemysłowej w typie **double**.

### 3.5 Dziedziczenie wielobazowe

W programie występuje dziedziczenie wielobazowe. Klasy `School` oraz `Hospital` dziedziczą po klasach `Special_element` oraz `Infrastructure`.

### 3.6 Niestandardowe biblioteki

Aplikacja posiada tekstowy interfejs użytkownika typu TUI. Wykorzystano niestandardową bibliotekę `conio.h`, w szczególności funkcję `getch()`. Funkcja jest wywoływana w pętli zmieniającej stan programu. W celu ograniczenia szybkości pracy pętli skorzystano z funkcji `Sleep(int)` z niestandardowej biblioteki `windows.h`.

## 4 Specyfikacja zewnętrzna

Program jest uruchamiany w linii poleceń.

## 5 Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem programowania obiektowego. W programie rozdzielono interfejs (komunikację z użytkownikiem) od logiki aplikacji (dane na temat miasta).

### 5.1 Ogólna struktura programu

W funkcji głównej wywołana jest funkcja view, w której jest wywołane główne menu. Z głównego menu można wykonywać akcje lub przejść do kolejnych menu.

### 5.2 Szczegółowy opis typów i funkcji

Szczegółowy opis typów i funkcji zawarty jest w załączniku.

## 6 Testowanie

Program został przetestowany poprzez używanie symulatora. Problemem jest brak sprawdzania poprawności wejść. Gdy wpiszemy litery kiedy program oczekuje cyfry aplikacja może nieprawidłowo zakończyć pracę.

## 7 Wnioski

Symulator miasta jest programem niezawierającym żadnego skomplikowanego algorytmu. Główną trudnością jest ilość elementów w mieście i związane z nimi właściwości. Symulator jest programem otwartym do dalszego rozwoju: można dodawać następne budynki, lub zamiast w kosoli przedstawiać miasto okienkowo.