

# **Przetwarzanie Równoległe i Rozproszone**

**Materiały do zajęć laboratoryjnych**



**Wyższa Szkoła  
Przedsiębiorczości  
i Administracji**

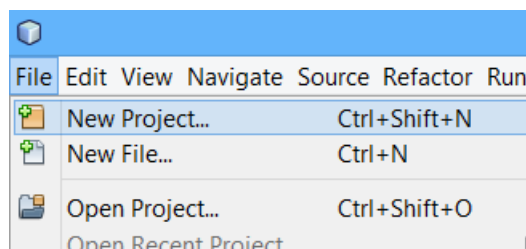
**mgr inż.**

**Kamil Zieliński**

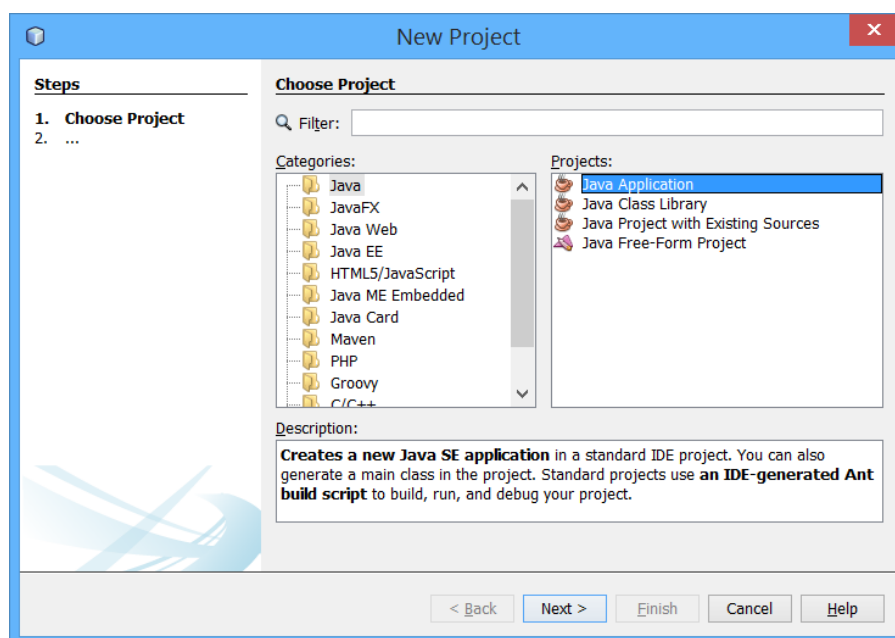
**Wyższa Szkoła Przedsiębiorczości i Administracji w Lublinie**

**2017/2018**

Celem laboratorium jest przygotowanie aplikacji wielowątkowej, symulującej problem wąskiego mostu. Aplikacja powinna zostać napisana w języku Java. Środowiskiem pracy powinien być NetBeans (lub inna platforma programistyczna). Instrukcje, zawarte w tym pliku, dotyczą środowiska NetBeans.



Pierwszym krokiem jest utworzenie nowego projektu. Odbywa się to z wykorzystaniem menu File / New Project, lub poprzez skrót klawiaturowy Ctrl + Shift + N.

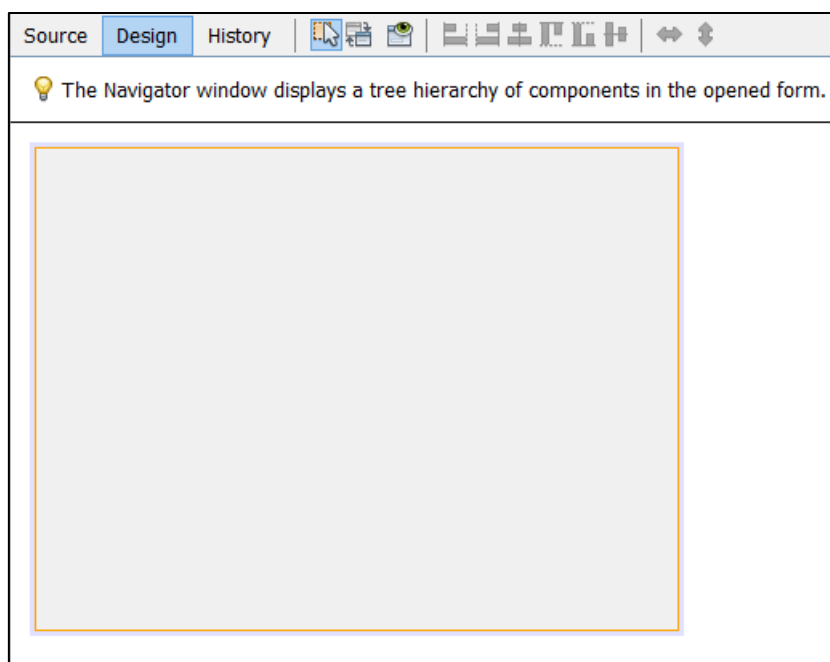


W wyświetlonym oknie należy wybrać opcję Java Application i kliknąć Next.

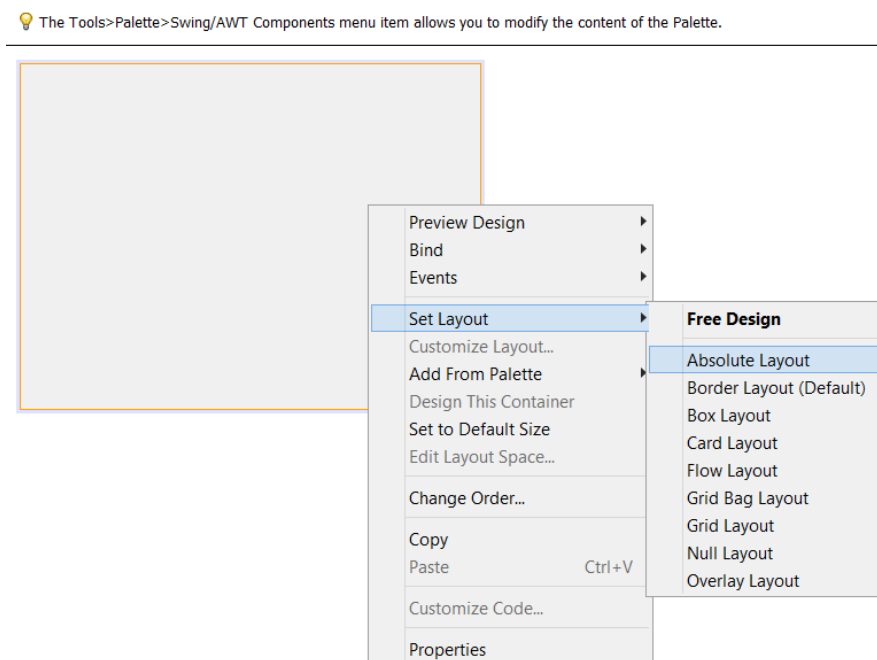
Kolejne wyświetlone okno, przeznaczone jest do wprowadzenia nazwy projektu oraz klasy main. W tym przypadku nie należy zaznaczać opcji *Create Main Class*. Utworzony zostanie projekt, nie posiadający żadnej klasy.

Po utworzeniu nowego projektu, konieczne jest dodanie okna JFrame. Odbyna się to poprzez kliknięcie prawym przyciskiem myszy na nazwę projektu (w sekcji Projects)

W oknie należy wprowadzić nazwę tworzonej klasy (okna) oraz kliknąć na przycisk Finish. Spowoduje to wyświetlenie nowego okna. Przyciski *Source* oraz *Design* przeznaczone są do przełączania się pomiędzy widokiem kodu Java oraz projektu okna.



Przed przystąpieniem do układania kontrolek na wyświetlonej formatce, należy dodać do niej layout. Odbyna się to z wykorzystaniem menu kontekstowego, dostępnego po kliknięciu prawym przyciskiem myszy na formatkę.



Po dodaniu layoutu do formatki, można rozpocząć rozmieszczanie na niej kontrolek.

Ogólną postać struktury kodu zawierającego wątek, przedstawiono poniżej.

```
public Thread exampleT = new Thread(new Runnable() {
    public void run(){
        while (true){
            SwingUtilities.invokeLater(new Runnable() {
                public void run() {
//metody wykonywane w wątku
                }
            });
            try {

                exampleT.sleep(1000);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

        }
    }
});
```

Zadanie polega na napisaniu aplikacji, składającej się z trzech wątków:

- Most
- Wschód
- Zachód

GUI aplikacji powinno zawierać kontrolki:

- Symbolizujące kolor semafora dla obu kierunków (można zrealizować to poprzez wykorzystanie kontrolki JLabel)
- Wyświetlające liczbę pojazdów po obu stronach mostu
- Przyciski pozwalające na dodanie 10 pojazdów do kolejki po każdej ze stron
- Przycisk, combobox oraz textbox. Po kliknięciu na przycisk, określona (w textbox) liczba pojazdów powinna zostać dodana do kolejki po stronie, wybranej w combobox.

Wątek most powinien implementować metody, odpowiadające za odczyt danych z pliku tekstowego, ustawianie semafora, zmianę kolorów świateł oraz zapis danych do pliku.

Plik tekstowy powinien składać się z 3 linii

- Liczba pojazdów na wschodzie
- Liczba pojazdów na zachodzie
- Semafor

Wątki Wschód oraz Zachód powinny sprawdzać ustawienie semafora oraz przepuszczać pojazdy, gdy otrzymają zezwolenie.

Warunki dodatkowe:

- Wykonanie przycisków, pozwalających na dodanie 10 pojazdów do kolejki na wschodzie i na zachodzie.
- Brak nadmiarowego kodu – w przypadku wykonywania tej samej operacji w kilku miejscach, powinna zostać ona ujęta w funkcję oraz wywoływana z odpowiednimi parametrami.
- Możliwość globalnej modyfikacji parametrów takich jak ścieżka do pliku, nazwa pliku. Zadanie to należy wykonać poprzez skorzystanie ze stałych.
- Zabezpieczenie kodu przed wystąpieniem wyjątków.
- Dodanie przycisku, który pozwoli na dodanie dowolnej liczby pojazdów (wprowadzonej w polu textbox) do wybranej strony mostu (wybór w polu combobox).
- Utrzymanie konwencji nazewnictwa kontrolek (jLblEastVehicleCount, jBtnAddVehicleToEast).
- Utrzymanie konwencji nazewnictwa w kodzie aplikacji
  - `public class MyFirstClass{ }` – nazwa klasy
  - `public void getMyFirstFunction()` – nazwa metody
  - `public int myFirstIntegerVariable` – nazwa zmiennej
  - `public final String MY_FIRST_STRING_CONSTANT` – nazwa stałej