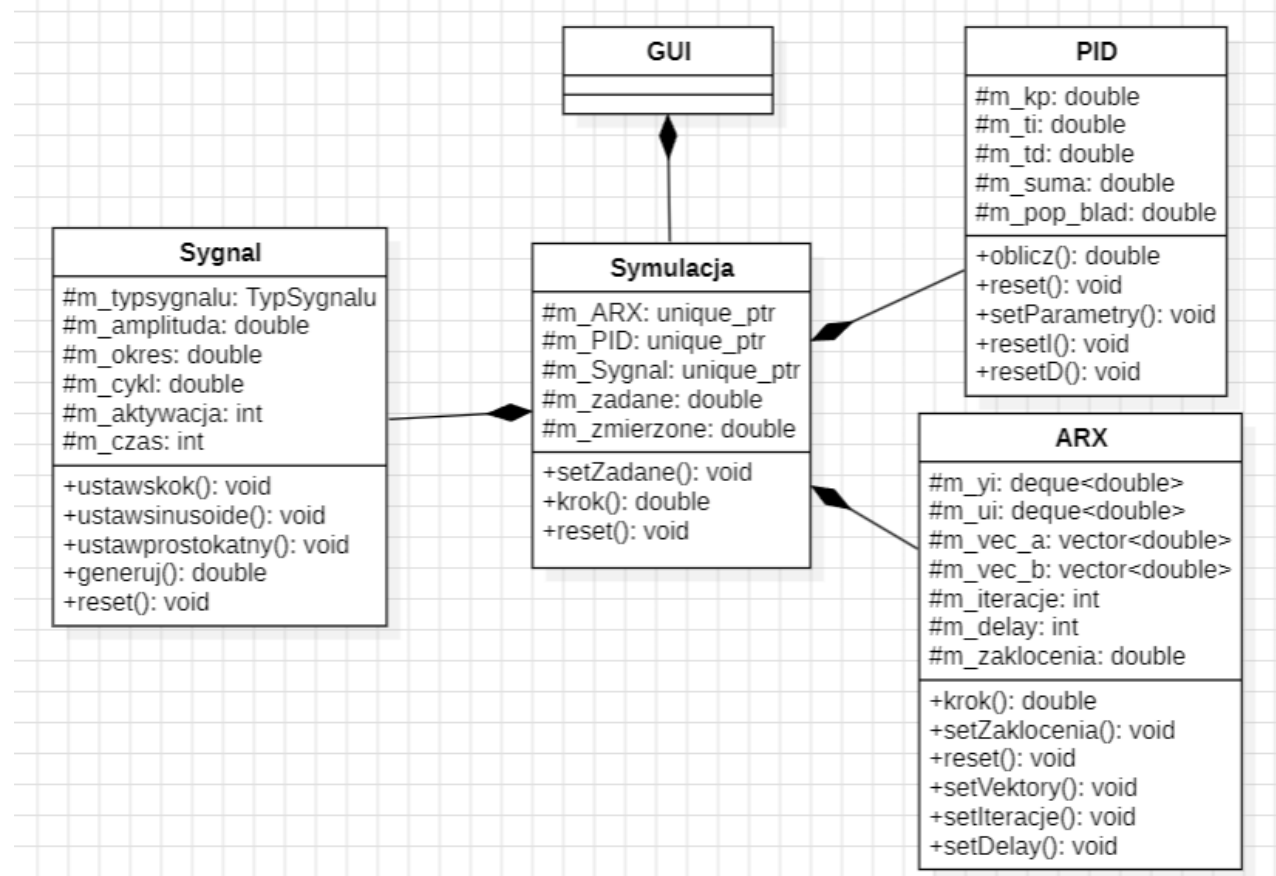


PROJEKT UAR

Mateusz Czernecki, Jakub Gondzik

Schemat UML – po zmianach

- GUI
- Symulacja (-setery pid i arx)
- PID (+ seter parametrów +reset I i D)
- ARX (+zakłócenia, -zadawanie wejścia, -metody które okazały się niepotrzebne)
- Zmiana relacji symulacji z GUI z agregacji na kompozycje



Wyróżnione funkcjonalności

Najbardziej satysfakcjonująca funkcjonalność

- Symulacja

```
void Symulacja::setZadane(double zadane)
{
    m_zadane = zadane;
}

double Symulacja::krok()
{
    if (!m_ARX || !m_PID || !m_sygnal)
    {
        throw std::logic_error("Nie ustawiono ARX, PID lub sygnalu");
    }

    double sygnal = m_PID->oblicz(m_zadane, m_zmierzone);
    m_zmierzone = m_ARX->krok(sygnal);
    return m_zmierzone;
}

void Symulacja::reset()
{
    m_PID->reset();
    m_ARX->reset();
    m_zmierzone = 0.0;
}
```

Najbardziej kłopotliwa funkcjonalność

- ARX

```
double ARX::krok(double u) //inna wersja kroku
{
    m_ui.push_back(u);
    if (m_ui.size() > m_vec_b.size())
    {
        m_ui.pop_front();
    }

    double wyjście = 0.0;

    for (size_t i = 0; i < m_vec_b.size(); i++)
    {
        wyjście += m_vec_b[i] * m_ui[m_ui.size() - 1 - i];
    }

    for (size_t i = 0; i < m_vec_a.size(); i++)
    {
        wyjście -= m_vec_a[i] * m_yi[m_yi.size() - 1 - i];
    }

    if (m_zaklocenia > 0.0)
    {
        wyjście += gzaklocen(generator);
    }

    m_yi.push_back(wyjście);
    if (m_yi.size() > m_vec_a.size())
    {
        m_yi.pop_front();
    }

    return wyjście;
}
```

Dziękujemy za uwagę