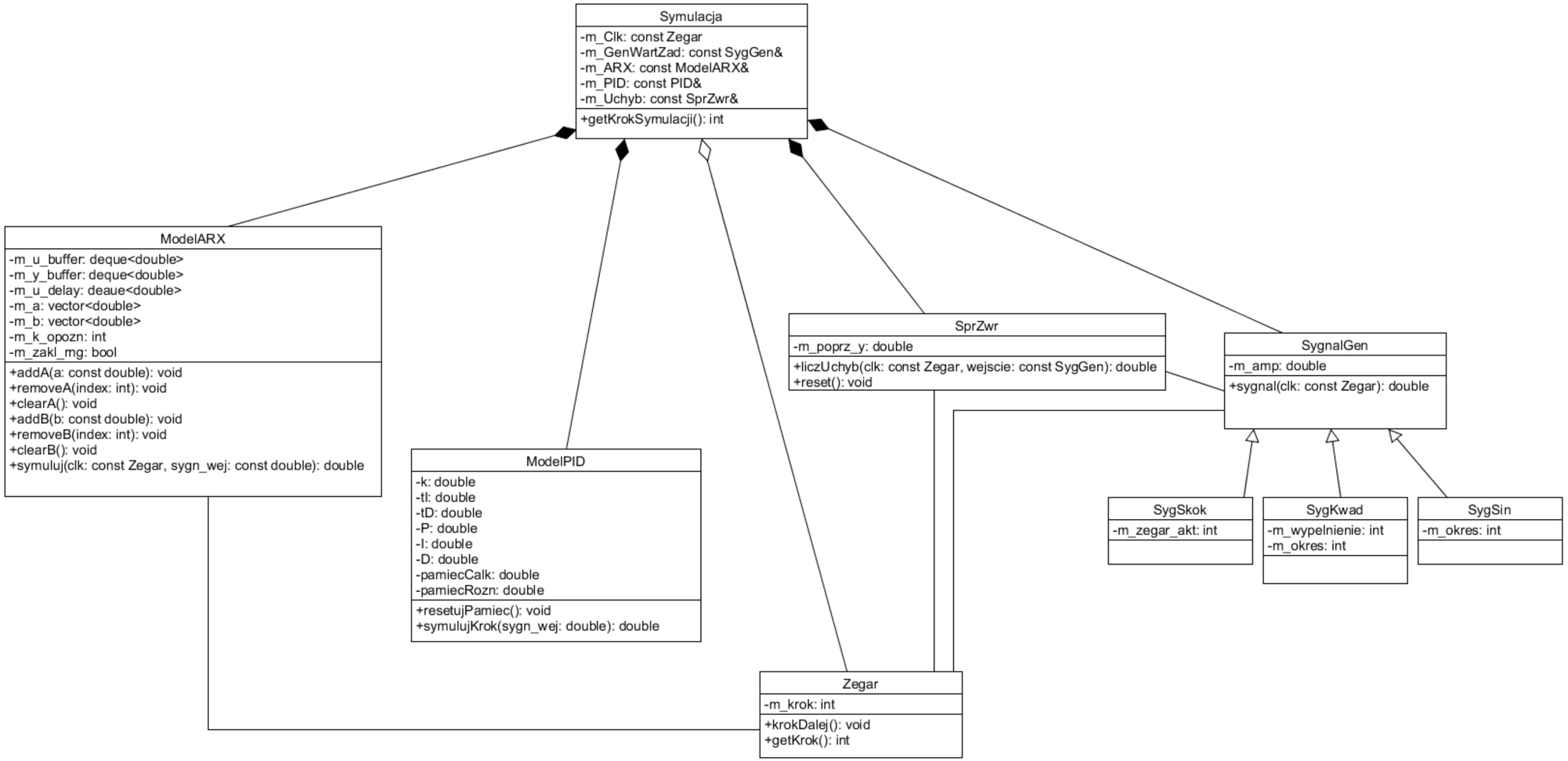
Politechnika Śląska

UCZELNIA BADAWCZA
INICJATYWA DOSKONAŁOŚCI

SYMULATOR UKŁADU AUTOMATYCZNEJ REGULACJI

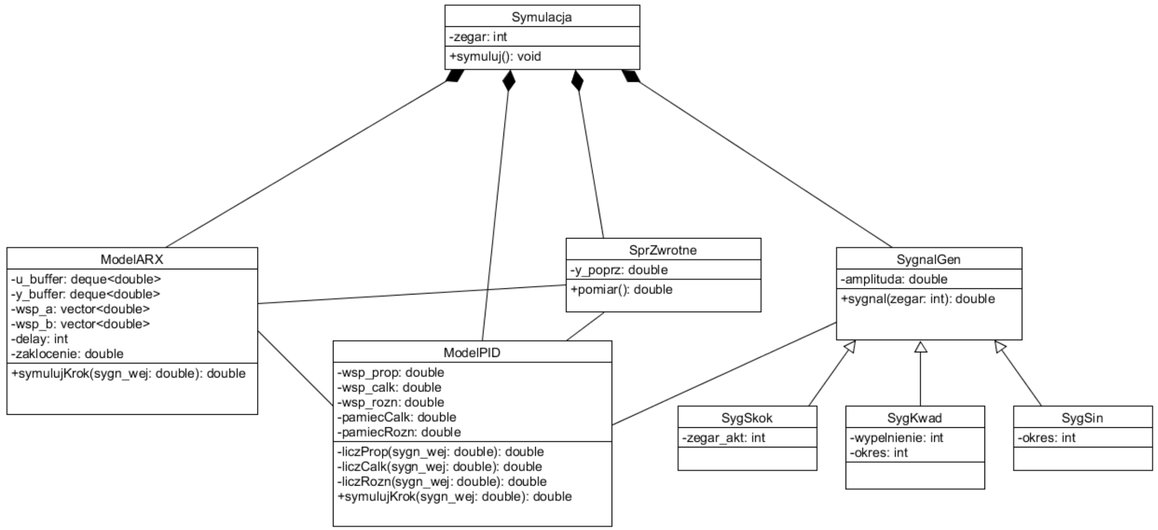
Michał Jaworek
Mateusz Rymoniak

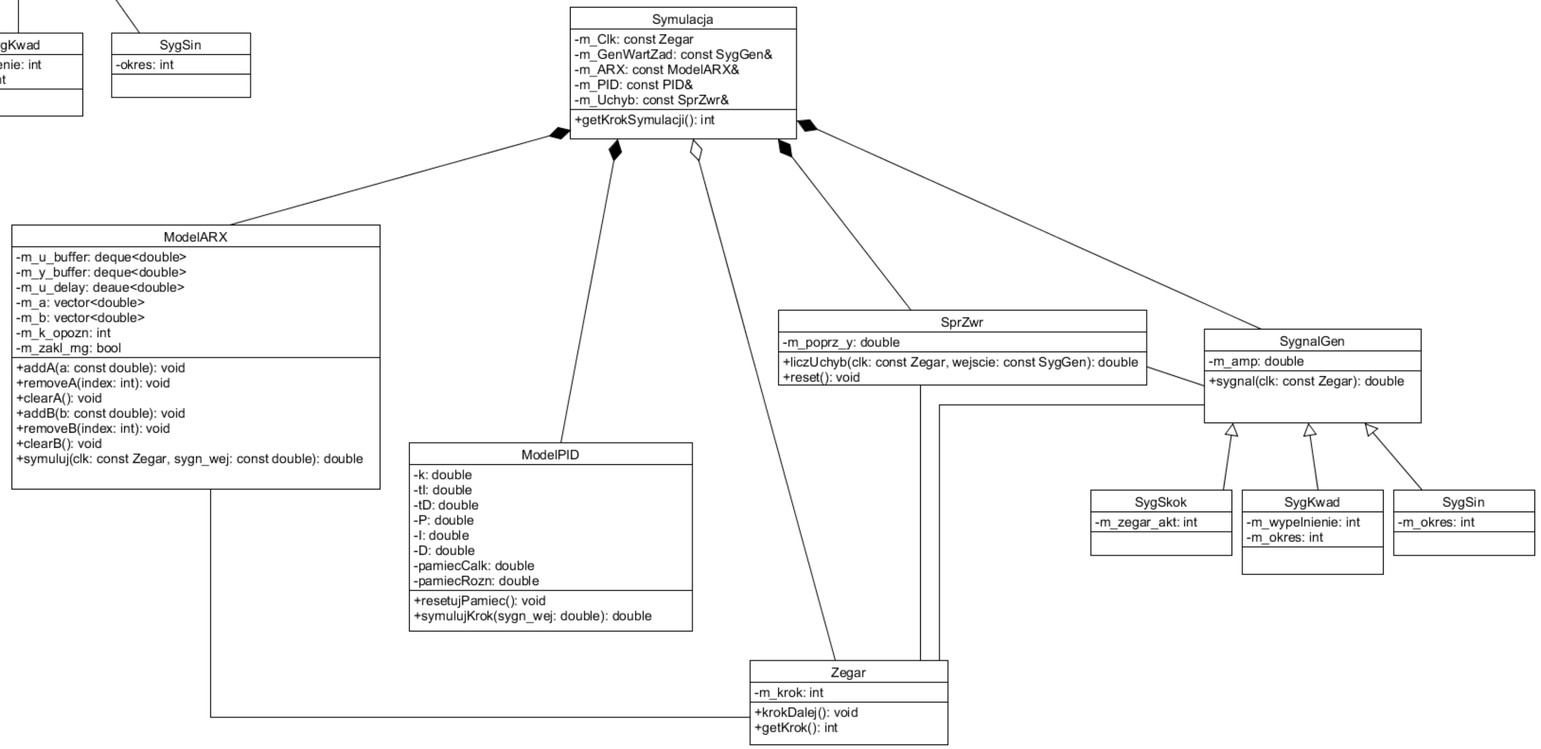
Schemat UML



Politechnika Śląska

Zmiany wstępnego projektu





Politechnika Śląska

Funkcjonalność, która sprawiła najwięcej satysfakcji – implementacja regulatora PID

pid.h — symulujKrokPID(double): double

```h
#ifndef PID_H
#define PID_H

class PID
{
    double k;
    double tI;
    double tD;
    double P;
    double I;
    double D;
    double pamiecCalk;
    double pamiecRozn;
    double getProp();
    double getCalk();
    double getRozn();
public:
    PID(double tempK, double tempTI, double tempTD);
    ~PID();
    void ustawK(double tempK);
    void ustawTI(double tempTI);
    void ustawTD(double tempTD);
    void resetujPamiecCalk();
    void resetujPamiecRozn();
    void resetujPamiec();
    double czescProp(double eI);
    double czescCalk(double eI);
    double czescRozn(double eI);
    double symulujKrokPID(double eI);
};

#endif // PID_H
```

pid.cpp* — pid::symulujKrokPID(double): double — CRLF — Linia: 76, kolumna

```cpp
#include "pid.h"

pid::pid(double tempK, double tempTI, double tempTD) : k{tempK}, tI{tempTI}, tD{tempTD}{

}

pid::~pid(){

}

void pid::ustawK(double tempK){
    k = tempK;
}

void pid::ustawTI(double tempTI){
    tI = tempTI;
}

void pid::ustawTD(double tempTD){
    tD = tempTD;
}

void pid::resetujPamiecCalk(){
    pamiecCalk = 0;
}

void pid::resetujPamiecRozn(){
    pamiecRozn = 0;
}

void pid::resetujPamiec(){
    resetujPamiecCalk();
    resetujPamiecRozn();
}

double pid::czescProp(double eI){
    return k * eI;
}

double pid::czescCalk(double eI){
    pamiecCalk += eI;
    return pamiecCalk / tI;
}

double pid::czescRozn(double eI){
    if (pamiecRozn == 0.0){
        pamiecRozn += eI;
        return 0.0;
    }else{
        double temp = tD * (eI - pamiecRozn);
        pamiecRozn = eI;
        return temp;
    }
}

double pid::getProp(){
    return P;
}

double pid::getCalk(){
    return I;
}

double pid::getRozn(){
    return D;
}

double pid::symulujKrokPID(double eI){
    P = czescProp(eI);
    I = czescCalk(eI);
    D = czescRozn(eI);
    return P + I + D;
}
```

Politechnika Śląska

Funkcjonalność, która sprawiła najwięcej kłopotów – implementacja zegara

zegar.h — getKrok() const: int

```h
#ifndef ZEGAR_H
#define ZEGAR_H

class Zegar
{
    int m_krok;
public:
    Zegar();
    void krokDalej();
    int getKrok() const { return m_krok; }
};

#endif // ZEGAR_H
```

zegar.cpp

```cpp
#include "zegar.h"

Zegar::Zegar()
    : m_krok{ 0 }
{}

void Zegar::krokDalej()
{
    m_krok++;
}

```