

Informatyka przemysłowa profil praktyczny
Semestr 3
30.01.2025r.

Projekt z przedmiotu „Programowanie komputerów” pt.

Symulator układu automatycznej regulacji (UAR)

Sekcja 2 Skład sekcji:
Szkółka Konrad
Michał Witczak

1) Podział obowiązków:

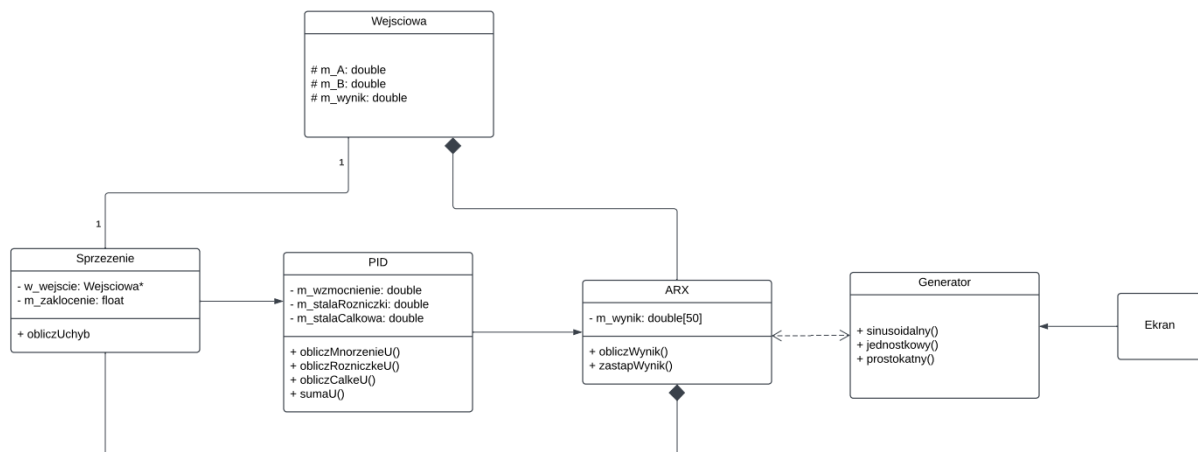
- Michał Witczak był w głównej mierze odpowiedzialny za kod backend, klasy regulatora PID, model ARX, klasę sprzężenia zwrotnego oraz generatora wartości zadanej, jako bardziej doświadczony i zaawansowany programista w zespole czuwał nad tym aby kod się kompilował i służył radą w każdej nawet najbardziej skomplikowanej sytuacji.

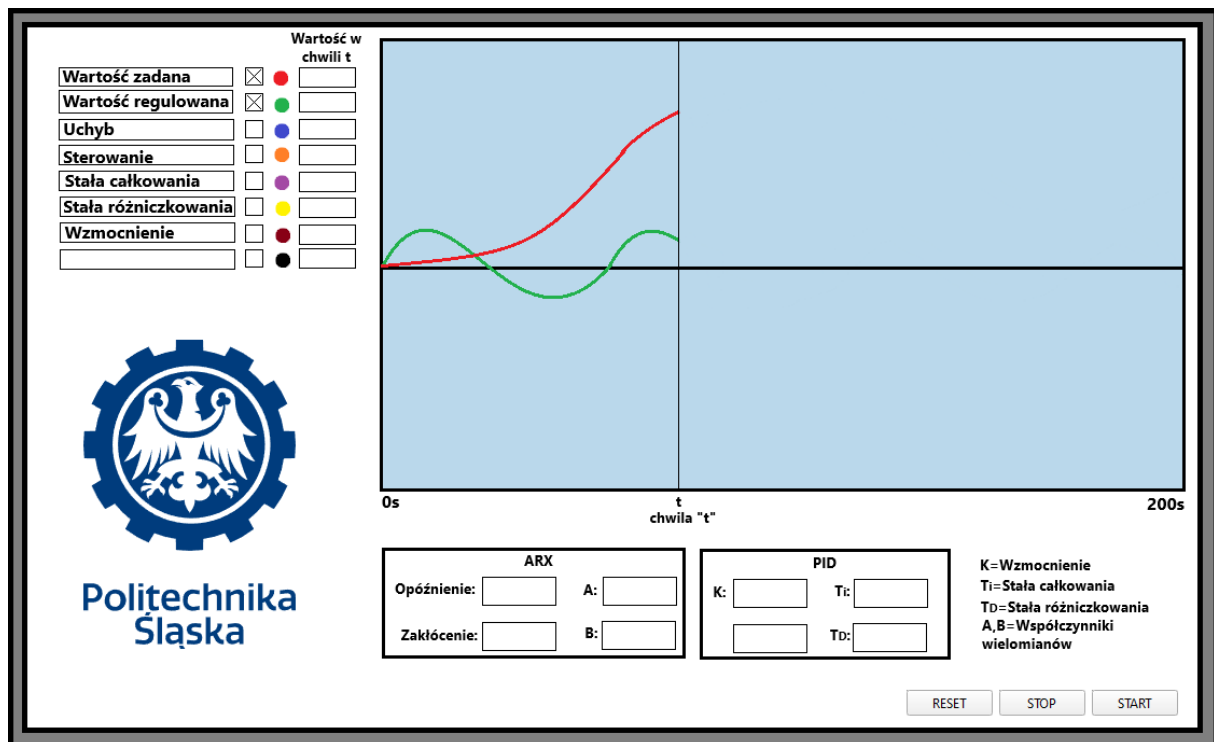
- Szkółka Konrad, odpowiedzialny w większej mierze za część merytoryczną i kreatywną taką jak prezentacje, sprawozdania, oraz wstępny i późniejsze projekty GUI oraz komunikacja z backendem oraz obsługę wykresów.

- W trakcie prowadzenia prac nad projektem obaj pisaliśmy kod backend jak i frontend według wcześniej ustalonego luźnego podziału jedna osoba pisała funkcjonalność wysyłała kod do repozytorium drugo osoba testowała kod później pisała swoją część i odsyłała pierwszej osobie do przetestowania i kontynuowania prac, miało to na celu zasymulowania pracy w oddalonym od siebie zespole korzystającym z systemu kontroli wersji GitHub oraz zmierzaniem się z towarzyszącymi temu procesowi problemami, ostatecznie jednak przez nasze niewystarczające zrozumienie tematu i spowodowany ową niekompetencją poślizg projektu doprowadził do przebudowy naszego sposobu pracy diametralnie, od ujednolicenia zadań tzn. Michał stał się głównym programistą piszącym większość kodu i czuwającym nad tym aby nie rodziło się więcej błędów związanych z programowaniem, Konrad natomiast pracował nad wyglądem i sposobem działania GUI gdyż obaj dopiero zaczynaliśmy przygodę z Qtcreatorem więc nauka od podstaw była naturalnym krokiem aby projekt mógł powstawać, prezentacje i sprawozdanie to również jego zasługa.

2) Historia projektu:

- Pierwszy UML oraz pierwsze GUI

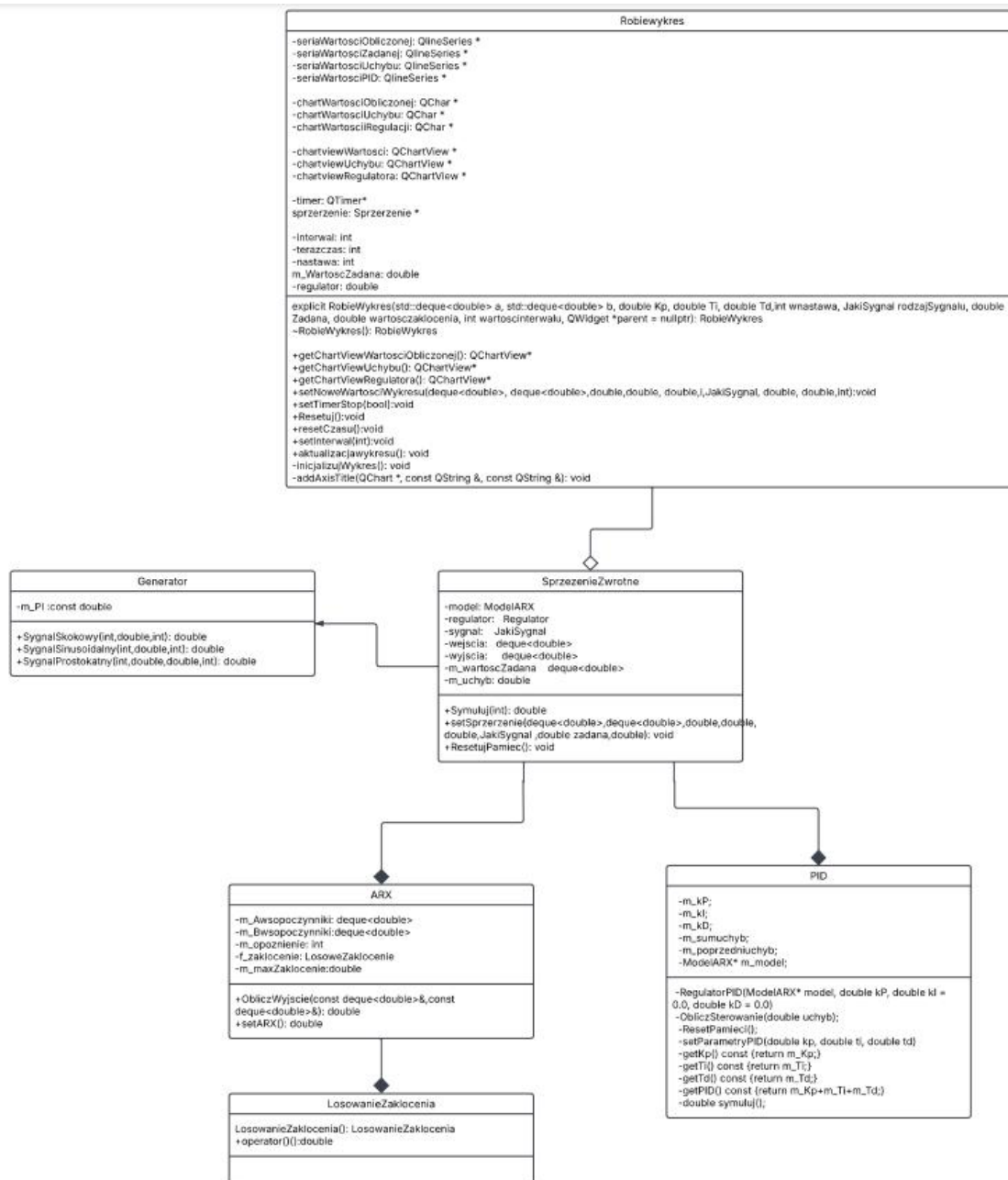




- Jak można zauważyć pierwszy projekt UML jak i GUI nie były zbyt przemyślane co mogło już wtedy świadczyć o niewystarczającym zrozumieniu założeń projektu.

- Na dzień dzisiejszy projekt GUI oraz UML wyglądają w następujący sposób:





- W schemacie UML głównymi zmianami jakie można zaobserwować jest logika całego drzewka zależności oraz bazowanie w głównej mierze na kompozycjach ponieważ są one prostsze w obsłudze i debugowaniu. Agregacja jest tylko w miejscu gdzie generujemy wykresy gdyż nie może istnieć bez całej logiki zawartej wcześniej, asocjacja obustronna między wykresami a zapisem ponieważ zapis musi znać stan wykresów, a wykresy powinny zaciągać dane startowe z zapisanych plików. Ostatecznie zależność sprzężenia zwrotnego od generatora.

- W kontekście GUI zmieniło się wiele od przejścia z jednego złożonego wykresu na 3 mniejsze bardziej czytelne po dodanie odpowiednich kontrolerek i przełączników po lewej stronie tak by obszar był ładnie zagospodarowany i nie trzeba go było załuszczać wypełniaczem.

3) Napotkane trudności:

- Początkowym i bardzo dużym problemem było poprawne zrozumienie przedstawionego problemu i treści zadania co skutkowało wieloma rebootami i zaczynaniem projektu od zera.

- Połączenie GUI z backendem też było nie lada wyzwaniem gdyż napisaliśmy program który działał w konsoli i wydawał się spełniać swoje zadanie obliczając kolejne wartości, połączenie tak napisanego programu z GUI spowodowało kolejne przebudowy w kodzie tak aby poprawnie działał z aplikacją a nie konsolą systemową.

- Łączenie z wykresami też było problematyczne ponieważ wprowadzenie niewielkich zmian potrafiło wywrócić działający wykres i doprowadzić go do błędnego pokazywania wartości a nawet znikania z ekranu w trakcie działania programu.

- Próbowaliśmy również zrobić zapis i odczyt do pliku niemniej zadanie to okazało się dla nas zbyt ambitne w porównaniu do czasu jaki mogliśmy mu poświęcić.

4) Czego nauczył nas projekt:

- Współpracy i pracy na cudzym kodzie - praca w parze nad projektem poprawiła naszą zdolność do współpracy począwszy od konsultowanych między sobą pomysłów po pisanie kodu gdzie niejednokrotnie dochodziło do pomyłek związanych z niezrozumieniem kodu napisanego przez drugą osobę co wymuszało na nas ciągłą komunikację i współpracę.

- Nieszablonowego myślenia - Nauczyła nas tego głównie praca nad GUI gdzie wielokrotnie musieliśmy szukać obejść i dziwnych rozwiązań, na przykład przy tworzeniu wykresów które od samego początku sprawiały wiele kłopotów.