Politechnika Wrocławska  
Wydział Informatyki i Telekomunikacji

**Projekt  
program symulujący przebieg studiów**

Prowadzący zajęcia:  
mgr inż. Damian Mroziński

Autorzy:  
Jakub Łazorko  
Kacper Jachimowski  
Informatyka Techniczna semestr II

# **Podstawowe informacje**

**Język programowania**: **C++**

**Github:** [**https://github.com/xololt-dev/OOP-Project**](https://github.com/xololt-dev/OOP-Project)

**Zakres projektu**: Projekt polega na symulacji przebiegu studiów w zależności od środowiska uczelnianego i predyspozycji studentów. Program podzielony jest na dwie główne części – **studenci** i **profesorowie**.

**Studenci** posiadają różne właściwości - ustalane na początku przez użytkownika, bądź losowe - takie jak: umiejętności z jakimi zaczynają studia, skłonność do imprezowania, skłonność do nauki, obecność na zajęciach i czy dzielą pracę ze studiami.

**Profesorowie** różnią się między sobą ilością kolokwium i ich poziomem trudności. Dodatkowo mamy możliwość ustawienia wartości edukacyjnej wygłaszanych przez nich wykładów. Tak jak w przypadku **studentów**, mamy opcję pełnego ustalenia parametrów przez użytkownika bądź pozostawienia ich jako losowe.

Program dodatkowo przewiduje możliwość wyznaczenia ilości semestrów wliczająca się w pojedynczą symulację studiów.

Moduł dodatkowy może zostać przeznaczony na wydarzenia losowe, przykładowo zajęcia stacjonarne lub zdalne, choroby, wariacje na zasadzie „gorszego/lepszego dnia”.

Wersja podstawowa symulacji przewiduje zakres **studentów** i **profesorów** w jednej **uczelni**. Jednakże jest opcja rozbudowy o część **uczelnianą**, przez co **studenci** będą mogli się przepisywać między **uczelniami**.

# **Główne założenia projektu:**

Program ma za zadanie przeprowadzić symulację przebiegu studiów dla grupy studentów o różnych predyspozycjach naukowych w określonym środowisku uczelnianym. Każda iteracja programu obejmuje jeden dzień z życia studenta z rozróżnieniem na poszczególne dni tygodnia oraz na tygodnie parzyste i nieparzyste. Symulacja kończy się po 7 semestrach, gdzie każdy semestr trwa sto pięć dni.

Program składa się  z klas głównych - odpowiedzialnych za studentów, prowadzących ćwiczenia oraz profesorów, oraz klasy poboczne pełniące funkcje interfejsów odpowiedzialnych za wczytywanie danych, przebieg dnia na uczelni i aktywności w czasie wolnym.

Studentom zaczynającym studia przypisywane są atrybuty takie jak umiejętności, skłonność od imprezowania, skłonność do nauki oraz informacje czy poza studiami jeszcze pracują. Oprócz tego student ma jeszcze zmienną określającą jego zmęczenie, które bezpośrednio przekłada się na jego obecność na zajęciach. Wykonywane przez nich zadania dzielą się na aktywności na uczelni oraz to jak spędzają swój czas wolny.

Informacjami jakie dostaje program o prowadzących zajęcia tu określanych profesorami są trudność prowadzonego przez nich kursu, ilość kolokwiów w trakcie kursu, wartość edukacyjna przekazywana studentom podczas zajęć. Każdy z profesorów prowadzi wykład w ustalonym dniu. W porównaniu do profesora prowadzący ćwiczenia może mieć jedne lub kilka zajęć tygodniowo.

Na podstawie dni zajęć prowadzonych dla każdego studenta przygotowywany jest plan według którego realizowany jest tydzień semestru. Student może mieć zarówno kilka godzin dziennie jak i mieć dzień wolny.

Funkcja realizująca zadania na dany dzień, sprawdza stan zmęczenia studenta i na jego podstawie określa czy dany student pójdzie na zajęcia czy nie. Jeśli zostanie zdeterminowane, że stan zmęczenia studenta jest niski i student może iść na zajęcia, uruchomiona zostanie funkcja która zwiększy poziom wiedzy studenta zależnie od prowadzącego zajęcia na które poszedł student oraz zwiększy zmęczenie studenta. W przypadku kilku zajęć w ciągu jednego dnia, stan zmęczenia sprawdzany jest przed każdymi zajęciami.

Wyjątkowymi przypadkami zajęć są te, na których występuje kolokwium. W jednym semestrze odbywa się odgórnie ustalona ilość kolokwiów, które służą za „checkpoint”, w którym sprawdzamy obecną wiedzę studenta. Jeśli jest ona nie wystarczająca, student nie jest wstanie otrzymać dyplomu (inaczej - ukończyć studiów).

Drugim przypadkiem, w którym student może nie ukończyć studiów jest oblanie egzaminu. Egzaminy zawsze odbywają się na końcu semestru.

Poza zajęciami na uczelni student może wykorzystać swój czas wolny po zajęciach na kilka możliwych sposobów. Mają one na celu podniesienia o niewielką wartość poziomu wiedzy bez obciążania zmęczenia oraz przede wszystkim zmniejszania poziomu zmęczenia. Dobrym sposobem na odpoczynek jest wyjście na piwo. Zmęczenie zmniejszane jest  również przy każdym cyklu dziennym jako “pasywny bonus” wynikający ze snu. Bonus ten jednak nie przysługuje gdy student wybierze się na imprezę.

**Uruchamianie programu:**

Aby poprawnie uruchomić program musimy wprowadzić 3 parametry podczas uruchamiania.

1. Nazwa pliku z danymi studentów, profesorów i ćwiczeniowców.
2. Nazwa pliku z dodatkowymi informacjami nt symulacji. Jeśli chcemy pracować na ustawieniach podstawowych należy w to pole wpisać dowolny znak alfabetu.
3. Nazwa pliku do którego chcemy zapisać wyniki symulacji.

Format każdej z nazw to nazwa\_pliku.txt.

# **Spis funkcji i klas programu:**

* **Afterhours**
* **Afterhours** - funkcja spajająca poniższe funkcje, zawiera losowanie szansy na realizację aktywności
* **isWeekend** - określa czy aktualny dzień to weekend, zależy od tego ile razy zostanie zrealizowana aktywność losowa
* **getSleep** - realizuje bonus za sen
* **goRelax** - realizuje aktywność zmniejszającą zmęczenie
* **goStudy** - realizuje aktywność zwiększającą wiedzę
* **goParty** - realizuje aktywność imprezy, zwiększającą zmęczenie, obniżającą wiedzę  i anulującą bonus wynikający ze snu
* **Day**
* **isOdd** - sprawdza czy realizowany dzień należy do tygodnia parzystego czy nieparzystego
* **getLectures** - realizuje wykłady danego dnia
* **getExcercises** - realizuje ćwiczenia danego dnia
* **Save**
* **getObjectsInfo** - przeprowadza pobieranie danych z pliku nt obiektów obiętych symulacją
* **getSimulationInfo -** pobiera dane z pliku dotyczące szczegółowych parametrów symulacji
* **saveToFile** - przeprowadza zapis danych do pliku
* **Staff**
* **Lesson** - klasa odpowiadająca za zajęcia
* **showDay** - zwraca dzień zajęć
* **showOccurence** - zwraca tydzień zajęć (parz./nieparz.)
* **showKnowledgetoGain** - zwraca wiedzę jaką student pozyska z zajęć
* **showCurrLesson** - zwraca numer aktualnych zajęć
* **Exercise (Lesson)** - klasa pochodna, odpowiada za ćwiczenia
* **setEParameters** - przypisuje ćwiczeniom ich parametry
* **showETestAmount** - zwraca ilość testów realizowanych podczas ćwiczeń
* **Lecture (Lesson)** - klasa pochodna, odpowiada za wykłady
* **setLParameters** - przypisuje wykładom ich parametry
* **showLExamCheck** - zwraca informacje o egzaminie
* **Staff** - klasa odpowiedzialna za pracowników uczelni
* **showId** - zwraca ID pracownika
* **showFirstName** - zwraca imię pracownika
* **showLastName** - zwraca nazwisko pracownika
* **showCourseDiffficulty** - zwraca trudność kursu
* **Professor (Staff)** - klasa pochodna, odpowiada za wykładowców
* **setPParameters** - przypisuje parametry wykładowcom
* **Academic (Staff)** - klasa pochodna, odpowiada za ćwiczeniowców
* **setAParameters** - przypisuje parametry ćwiczeniowcom
* **Student**
* **Student** - klasa odpowiedzialna za studentów uczelni
* **setSParameters** - przypisuje parametry studentom
* **updateKnowledge** - aktualizuje poziom wiedzy studenta
* **updateFatigue** - aktualizuje poziom zmęczenia studenta
* **showSId** - zwraca ID studenta
* **showFatigue** - zwraca poziom zmęczenia studenta
* **showKnowledge** - zwraca poziom wiedzy studenta
* **showSemester** - zwraca semestr studiów
* **showWorking** - zwraca informację czy student pracuje
* **showStudying** - zwraca informację czy student dalej studiuje

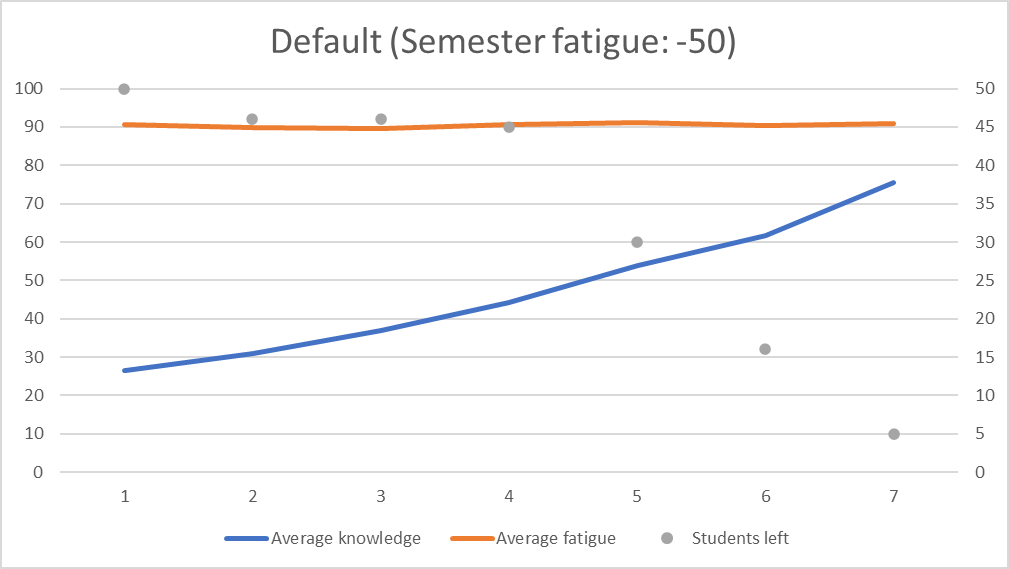
# **Diagram przypadków użycia**

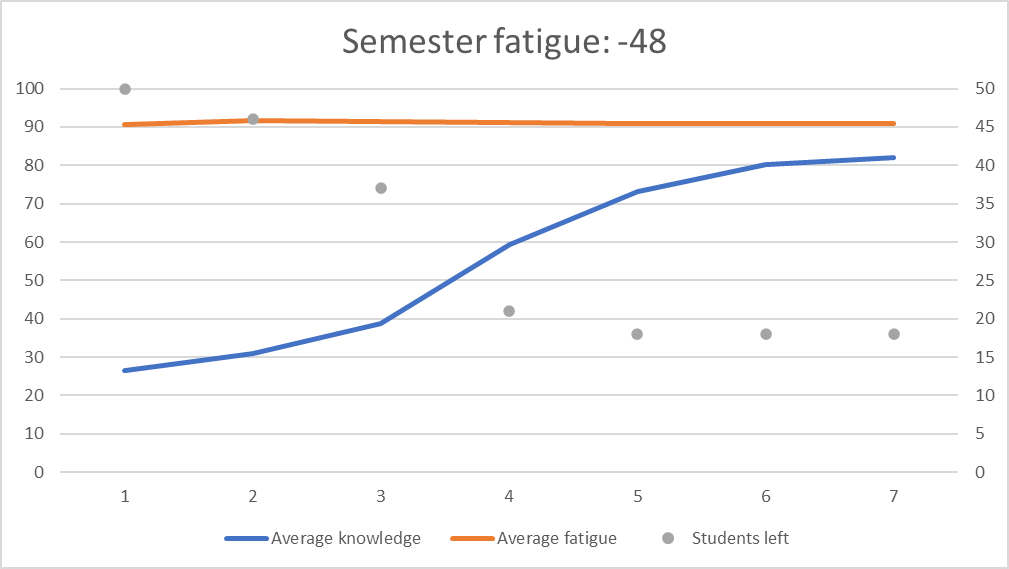
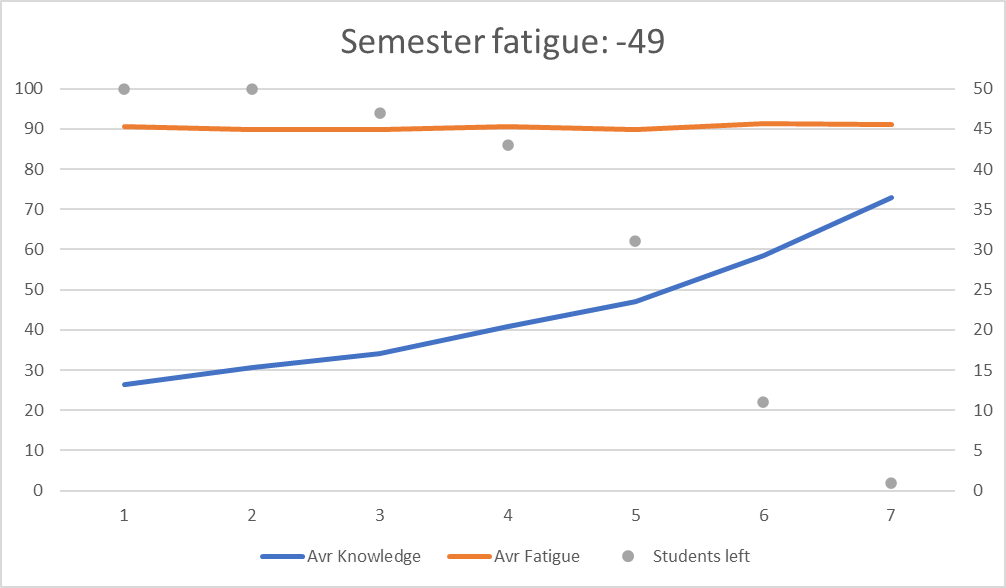
# **Diagram obiektów**

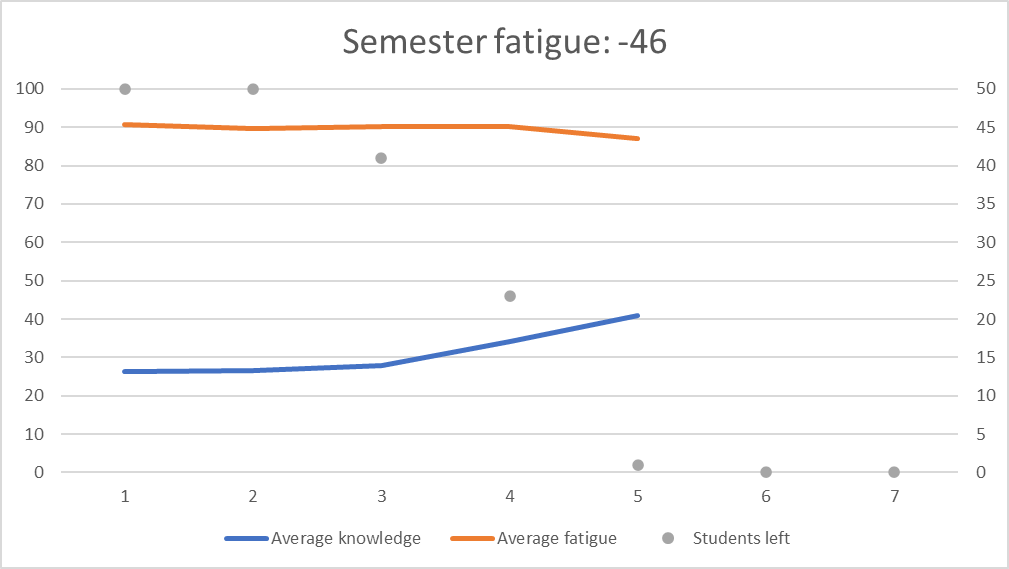
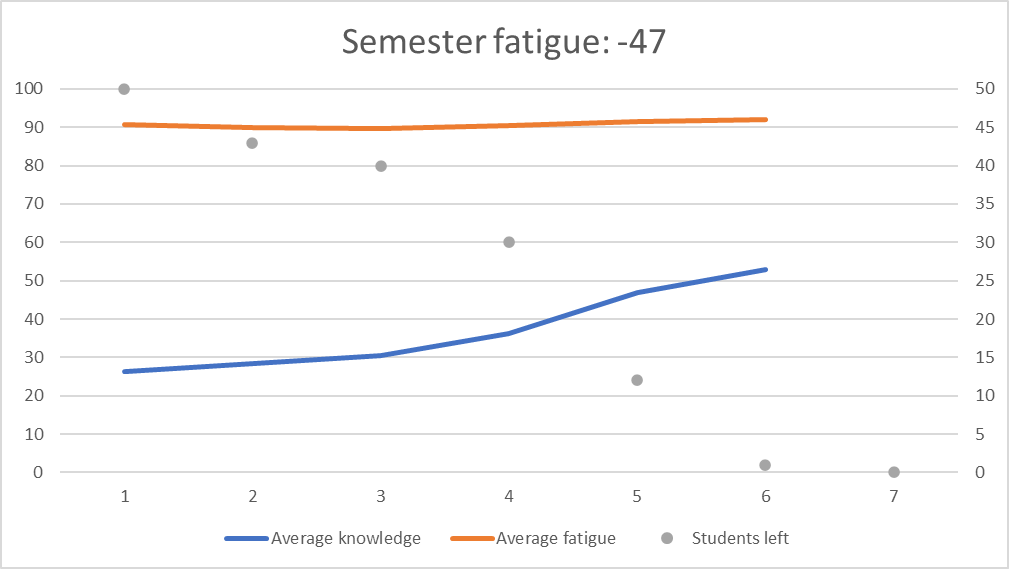
# **Analiza wpływu zmiennej na symulacje**

Za zmienną której wpływ na symulację będziemy rozważać przyjmujemy wartość o jaką obniżane jest zmęczenie studenta po zakończeniu semestru. Wynikami analizy będą liczba studentów, jaka pozostaje na uczelni, ich poziom zmęczenia oraz poziom wiedzy. Danymi wejściowymi było 50 studentów, 4 wykładowców i 4 ćwiczeniowców, ich dane początkowe nie zmieniały się na przestrzeni analizy.

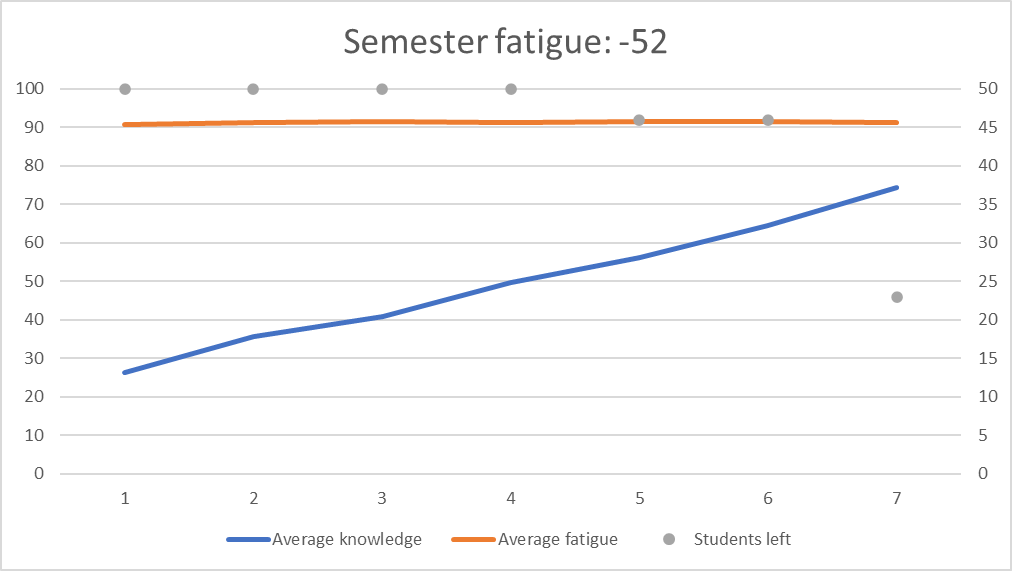
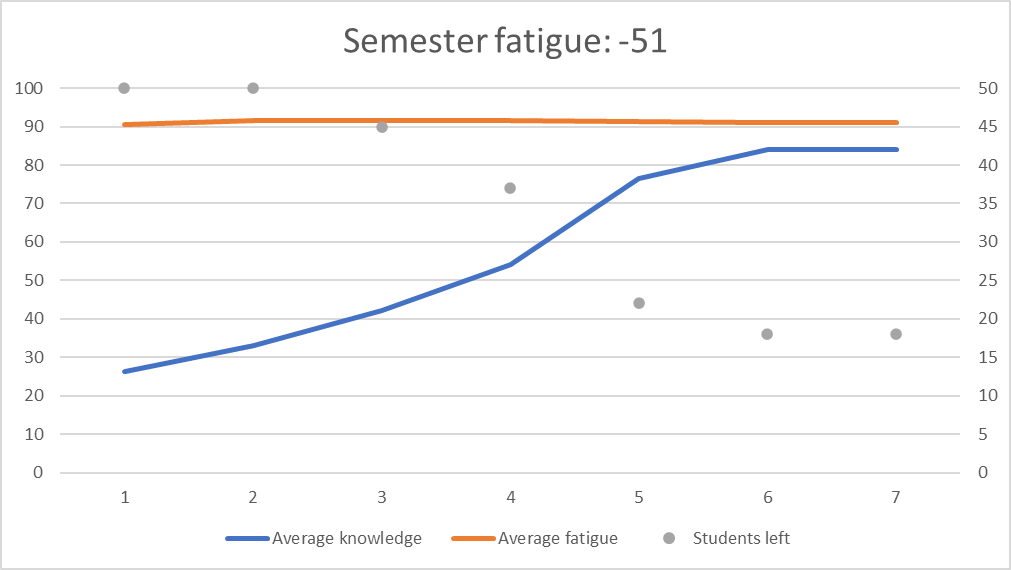
Za bazę przyjmujemy wartość obniżenia zmęczenia o 50 punktów

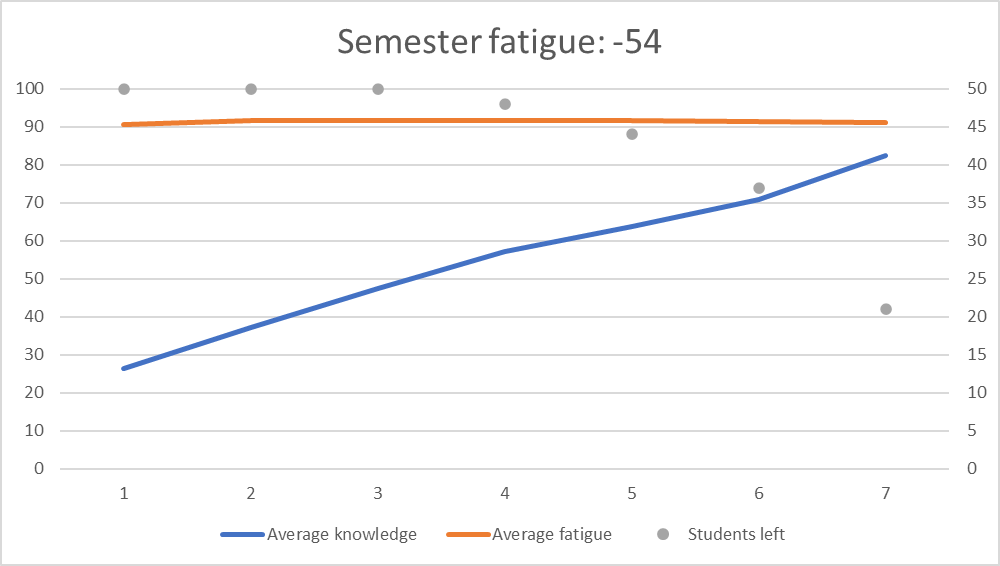
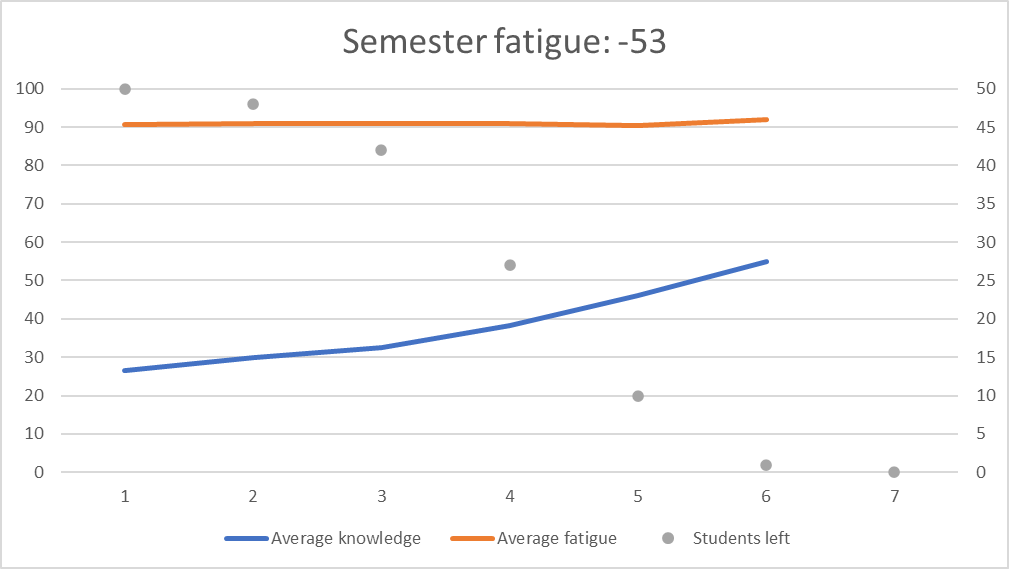
Dla wartości mniejszych niż bazowa wykresy przedstawiają się następująco:



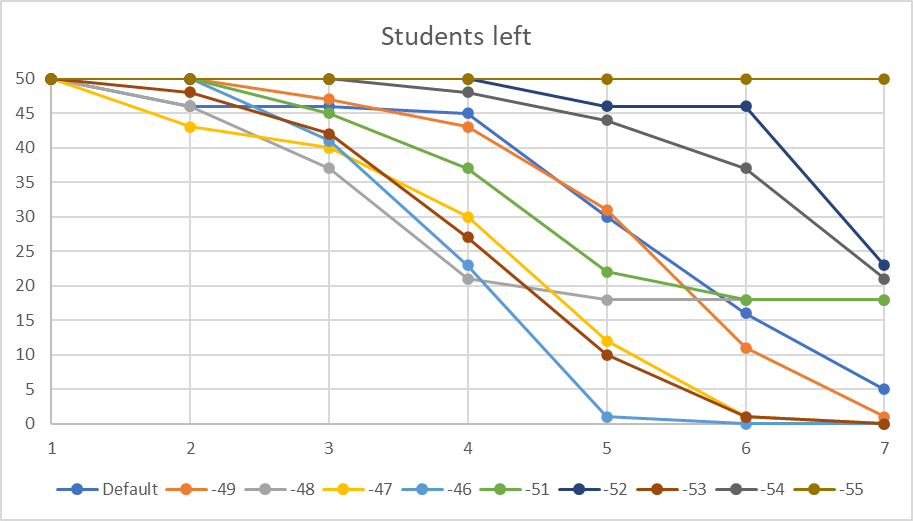


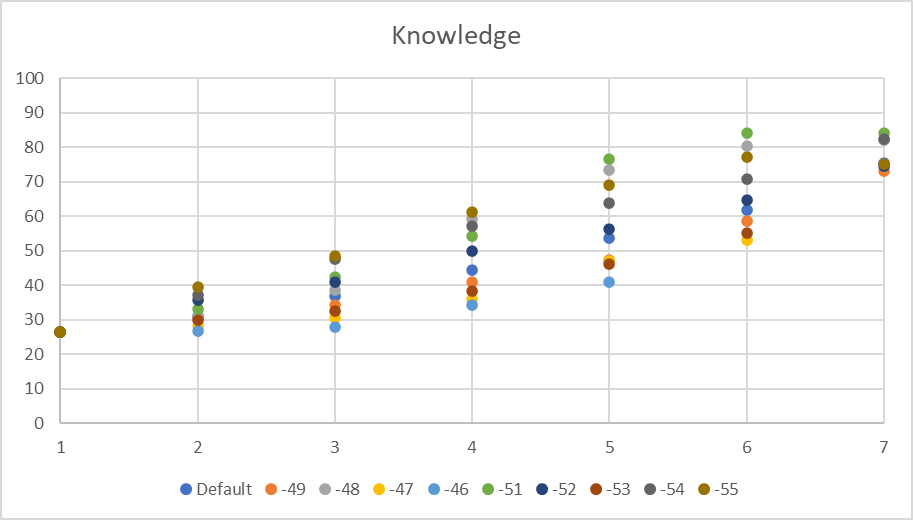
Dla wartości większych niż bazowa następująco:





Podsumowując:





**Wnioski wynikające z analizy**:

Poziom zmęczenia na koniec semestru za każdym razem jest wysoki i nie zależy od zmiennej.

Poziom wiedzy średnio jest tym większy im większa jest wartość zmiennej.

Liczba studentów jest średnio odwrotnie proporcjonalna do wartości zmiennej.

Dla zmiennej o wartości 47 po 6 semestrze nie zostaje żaden student. Taki sam wynik ale już po 5 semestrze występuje przy wartości zmiennej równej 46.

Przy wartości 55 żaden ze studentów nie odpada przez cały czas trwania studiów.

# **Rzeczy do poprawy:**

* wartości domyślne nie są zbalansowane pod względem zmienności wyników,
* zmiana zakresu niektórych danych, przykładowo „wiedza” – zwiększenie zakresu z 0 - 14 na semestr na większy zwiększyłoby różnorodność wyników,
* zmiana formatowania danych wyjściowych w celu ich łatwiejszej obróbki.

# **Możliwości rozbudowy:**

* dodanie wydarzeń losowych w celu zwiększenia realizmu,
* dodanie poprawy kolokwiów,
* dodanie możliwości przenoszenia między uczelniami.