

# Języki Programowania, wtorek 3.11.2020

## Zadanie 3

Ideą programu jest stworzenie prostej bazy “bazy danych” w C++ opisującej polski parlament. Polski Sejm liczy obecnie 460 posłów, zatem w programie stworzymy tablicę o rozmiarze [460] (można stworzyć tablicę statyczną, ale preferowane są tablice dynamiczne). Zakładamy, że każdy poseł ma: imię, nazwisko, przynależność partyjną, oraz wiek. Nowych posłów można dodawać do bazy danych. Można również wypisywać cały skład Sejmu. Program umożliwi również przeprowadzenie głosowania nad nową ustawą.

1) Stwórz klasę `Posel` zawierającą trzy składniki prywatne: imię (`std::string`), nazwisko (`std::string`), nazwę partii (`std::string`), oraz wynik głosowania (`int`). Klasa powinna być podzielona na dwa oddzielne pliki: `Posel.h` i `Posel.cpp`. W trzecim pliku powinna znajdować się funkcja główna `main`.

Klasa powinna zawierać metody:

- `UstawImieNazwisko(std::string imie, std::string nazwisko)` - ustawiająca imię i nazwisko posła
- `UstawPartia(std::string nazwa_partii)` – ustawiająca przynależność partyjną
- `UstawWynikGlosowania(int glos)` – ustawiająca wynik głosowania posła („0” - przeciw, „1” - za, „2” - wstrzymuję się)
- `Wypisz()` - wypisująca na ekran informacje o danym posle (wszystkie pola prywatne)

2) Ponadto, piszemy funkcję globalną zaprzyjaźnioną z klasą `Posel`:

`WynikGlosowania(Posel *sejm, int liczbaPoslow)` – zwracającą wartość „true” (lub „1”), jeżeli większość posłów z podanej liczby posłów jest na „tak”, lub „false” (lub „0”), jeżeli większość posłów z podanej liczby posłów jest na „nie” (zakładamy zwykłą większość niezbędną do przyjęcia ustawy – liczba głosów „za” musi być większa niż „przeciw”)

3) Należy przygotować „bazę danych” parlamentu z możliwością dodawania, usuwania oraz wypisywania posłów używając w tym celu tablicy (najlepiej dynamicznej) na 460 posłów. W głównym programie (funkcji `main`) wymagana jest pętla `while` zawierająca instrukcja typu `switch - case`, pytająca w pętli o podanie liczby od 1-4 lub 0.

- Wpisanie 1 - wypisanie wszystkich zapisanych posłów na ekran. (1 p)
- Wpisanie 2 - dodanie nowego posła do “bazy danych”. (1 p)
- Wpisanie 3 - zmiana przynależności partyjnej posła o konkretnym indeksie z „bazy danych” (0.5 p)
- Wpisanie 4 - Przeprowadzamy głosowanie nowej ustawy, np. o Trybunale Konstytucyjnym – za pomocą komendy `rand()` ustawiamy każdemu posłowi (z dostępnej liczby posłów) jego indywidualny wynik głosowania (0, 1 lub 2) i używamy funkcji zaprzyjaźnionej w celu wypisania informacji o głosowaniu (1.5 p)
- Wpisanie 0 - powinno przerwać pętlę oraz spowodować opuszczenie programu. (0.5 p)
- Wpisanie liczby innej niż dozwolone - program zakomunikuje, iż została wprowadzona błędna wartość. (0.5 p)

6) Program powinien być kompilowany przy pomocy komendy „make” (należy stworzyć odpowiedni Makefile!)

### Makefile

```
CXX = g++
```

```
CXXFLAGS = -Wall
```

```
LFLAGS =
```

```
OBJS = posel.o main.o
```

```
all: main
```

```
main: $(OBJS)
```

```
$(CXX) $(CXXFLAGS) $^ -o $@
```

```
clean:
```

```
rm -f *.o main
```

```
.PHONY: all clean
```