

Program to symulacja różnych lig piłkarskich składających się z 20 drużyn. Opis klas, funkcji i zmiennych w programie:

class Team- klasa reprezentująca drużyny w lidze

string name- nazwa drużyny

float quality- jakość danej drużyny

int points- punkty danej drużyny

int form- forma drużyny

void wykonajOperacje()- scala nazwę drużyny i przypisuje jej formę i punkty początkowo równe 0, wypisuje na ekran drużyny danej ligi i ich jakość

class Match-klasa reprezentująca symulacje meczu

static void Play(Team* Team1, Team* Team2)

```
{
    if(rand()%100<50+(Team1->quality+Team1->form-Team2->quality-Team2->form))
    {
        Team1->points+=3;
        Team1->form++;
        Team2->form--;
    }
    else if(rand()%100<50+(Team2->quality+Team2->form-Team1->quality-Team1->form))
    {
        Team2->points+=3;
        Team2->form++;
        Team1->form--;
    }

    else
    {
        Team1->points++;
        Team2->points++;
    }
}
```

Jeżeli wylosowana liczba jest mniejsza od różnicy jakości i form drużyn, punkty pierwszej drużyny są zwiększane o 3, jej forma o 1, a forma drużyny przeciwnej zmniejsza się o jeden (pierwsza drużyna wygrała mecz), w razie remisu obie drużyny dostają jeden punkt a ich forma nie zmienia się

class Season-klasa reprezentująca sezon danej ligi

```
static void Weeks (Team table[])
{
    for (int round = 0; round < 38; round++)
    {
        for (int match = 0; match < 10; match++)
        {
            int n = 20 - 1;
            int home = (round + match)%n;
            int away = (round - match + n)%n;
            if (match == 0)
                away = 20 - 1;
            if (round%2)
                std::swap(home, away);

            Match::Play(&table[home], &table[away]);
        }
    }
}
```

Generacja terminarzu rozgrywek i wywołanie funkcji play w celu rozegrania spotkania.

W lidze składającej się z 20 drużyn mamy 38 kolejek(**round**) w każdej kolejce jest 10 meczów(**match**)

```
std::ifstream plik;
```

```
plik.open( nazwaPliku.c_str() );
```

```
if( !plik.good() )
```

```
    return false;
```

```
Team tablica[20];
```

```
for(int i=0;i<20;i++)
```

```
{
```

```
    string a;
```

```
    std::string b;
```

```
    int c;
```

```
    plik >> a >> b >> c;
```

```
        tablica[i].wykonajOperacje( a, b, c );
```

```
}
```

Wczytywanie danych z wybranego wcześniej pliku i przekazanie ich do funkcji **void wykonajOperacje()** znajdującej się w klasie **Team**

```
cout << "po sezonie" << endl << endl;;
```

```
for (int i = 0; i < 20; i++)
```

```
    for(int j = 0; j < 20; j++)
```

```
        {
```

```
            if (tablica[i].points > tablica[j].points)
```

```
                swap(tablica[i], tablica[j]);
```

```
        }
```

```
for (int i = 0; i < 20;i++)
```

```
cout << tablica[i].name << " " << tablica[i].points << "pkt" << endl;
```

Sortowanie a następnie wyświetlanie drużyn po sezonie według ilości zdobytych punktów, w ten sposób wyświetla się nam końcowa tabela ligi