

# Konspekt do projektu pt. „Intel Threading Building Blocks”

## Zaawansowane Techniki Programowania

Autor: Karolina Mizera

Data wykonania: 3 grudnia 2018

### Zadanie 1.

Proszę zaimplementować funkcję `ParallelMonteCarlo`, która zwraca wyestymowaną wartość liczby PI dla N elementów zgodnie z metodą Monte Carlo. Funkcja `ParallelMonteCarlo` wykorzystuje algorytm `tbb::parallel_reduce`, dla którego argumentem funkcyjnym może być funktor albo lambda. Pliku `main.cpp` proszę nie modyfikować. Spodziewany(!) output programu podany jest na końcu pliku `main.cpp`.

Generator liczb pseudolosowych:

```
float RandomFloat(float min, float max)
{
    std::random_device rd;
    std::mt19937 gen(rd());
    std::uniform_real_distribution<> dis(min, max);
    return dis(gen);
}
```

### Zadanie 2

Przedmiotem zadania jest implementacja:

1. klasy szablonowej `Vector` opakowującej `tbb::concurrent_vector` oraz rozszerzająca go o metodę `print()`.
2. algorytmu `quick_sort` z wykorzystaniem zarządzania task'ami implementowanymi przez klasę `tbb::task_group`.

Proszę nie modyfikować pliku `main.cpp`. Spodziewany(!) output programu podany jest na końcu pliku `main.cpp`.

### Zadanie 3.

Proszę w pliku `main.cpp` zaimplementować graf, wykonujący podstawowe obliczenia na zadanej wejściowej: promień `r`.

Graf oblicza: obwód koła, powierzchnie koła, pole powierzchni kuli oraz objętość kuli.

Schemat implementowanego grafu znajduje się w pliku *graph.jpg* załączonym w katalogu zadania.

Ostatni węzeł w grafie jest źródłem output'u zamieszczonego na końcu pliku.

Graf należy napisać w pliku `main.cpp`.