

# Zasady kolokwium

## Przebieg:

- Czas przeznaczony na rozwiązanie kolokwium to 1 godzina 30 minut.
- Od momentu udostępnienia treści, przez 1,5 h aktywne będzie zadanie w systemie eKursy w którym należy umieścić rozwiązanie w formie plików tekstowych z rozszerzeniem .cpp/.h/.hpp.
- Nie ma preferencji co do organizacji kodu względem podziału na pliki - zadanie może być przesłane w jednym pliku .cpp lub podzielone na wiele plików w zależności od indywidualnych preferencji/wygody.
- Tylko rozwiązania umieszczone w wyznaczonym zadaniu na eKursy będą akceptowane, nie dopuszcza się innych form dostarczenia rozwiązania (wraz z przykładowym kolokwium udostępnione zostało zadanie na eKursy które działa tak samo jak to na właściwym kolokwium - proszę przeciwzyć wgrywanie odpowiedzi).

## Ocenianie:

- **Błędy kompilacji dyskwalifikują z dalszej oceny i skutkują wynikiem 0 pkt.**
- Pełna ocena za zadanie zakłada nieużywanie zmiennych globalnych oraz niekopiowanie argumentów tam, gdzie nie jest to konieczne (proszę korzystać z referencji i stałych referencji).
- Jeśli w treści są podane nazwy funkcji, klas itd. należy ściśle się ich trzymać.
- Unikaj tworzenia publicznych pól klas, staraj się odpowiednio przygotować interfejs klasy.

## Treść:

1. Stwórz klasę **Car**, która będzie zawierała pola: **name**, **max\_speed**, **driver\_name**.
2. Stwórz klasy **FamilyCar** i **F1Car**, wykorzystaj do tego mechanizm dziedziczenia i jako klasę bazową przyjmij Car. Rozszerz klasy o odpowiednio **number\_of\_seats** i **laps\_on\_fuel**.
3. Stwórz klasę **Race**, która będzie w sobie zawierała kolekcję samochodów(pole **cars**) oraz ilość okrążeń w wyścigu (**num\_of\_laps**). Wykorzystaj do tego wiedzę z zajęć. Uwaga! Klasa powinna zawierać tylko jedną kolekcję jako pole. Podpowiedź: wykorzystaj inteligentne wskaźniki.

4. Stwórz obiekt klasy **Race**, a następnie zaimplementuj metodę umożliwiającą dodawanie zarówno samochodów rodzinnych jak i F1.
5. Dodaj 5 samochodów rodzinnych i 2 bolidy F1. Ich nazwy powinny zawierać conajmniej 5 znaków. Rozszerz wszystkie klasy o odpowiednie konstruktory pozwalające ustawić wszystkie pola.
6. Rozszerz klasy o metodę **get\_score** przyjmującą jako argument ilość okrążeń. Dla samochodu rodzinnego powinna ona zwracać iloczyn `max_speed` \* ilość okrążeń. Dla bolidu F1 powinno to być `max_speed` \* (ilość okrążeń - ilość okrążeń / `laps_on_fuel`).
7. Rozszerz klasę Car o metodę **get\_full\_name** zwracającą trzy pierwsze znaki nazwy oraz nazwę kierowcy.
8. Do klasy Race dodaj metodę **play**, która będzie odpowiadała za wyścig. Jej logika powinna być następująca: na podstawie wszystkich wyników wszystkich obiektów w kolekcji (metoda `get_score`), uporządkuj samochody malejąco i wypisz ich nazwy (`get_full_name`) w konsoli. Wywołaj tą metodę na istniejącym obiekcie.
9. Do klasy Race metodę znajdującą samochód o najmniejszej maksymalnej prędkości. Wynik zaprezentuj w konsoli. Wywołaj tą metodę.
10. Do klasy Race metodę **get\_car\_name\_by\_driver**, która będzie szukała obiektu o odpowiedniej nazwie kierowcy i zwracała nazwę tego samochodu. Zaprezentuj jej działanie.

Powodzenia!