

PROGRAMOWANIE MOBILNE

LAB2 – Programowanie w Kotlinie, część I

1. Rozwiązań do poniższych zadań prześlij zbiorczo w jednym pliku pdf za pomocą platformy moodle <https://elf3.pk.edu.pl>. Po przesłaniu rozwiązań na platformę moodle poczekaj na prowadzącego, który zweryfikuje je bezpośrednio przed końcem zajęć, bądź przygotowany/a na demonstrację działania przesłanego rozwiązania. Pamiętaj, że weryfikacja zadań przez prowadzącego rozpocznie się 40 minut przed końcem zajęć i już wtedy Twoje odpowiedzi powinny znajdować się na platformie moodle.
2. Deklarowanie zmiennych.

Student może zdobyć zaliczenie z informatyki lub matematyki, dzięki czemu uzbiera odpowiednią ilość punktów aby uzyskać wpis na kolejny semestr. Należy zdefiniować zmienną *hasCredit*, która pozwoli przechowywać informacje czy student zdobył zaliczenie jednego z tych przedmiotów czy nie. Początkowy stan tej zmiennej ma oznaczać, że gracz zaliczenia nie uzyskał.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacja tego zadania, nie wgrywaj całego projektu.

3. Wnioskowanie typów.

Student posiada również inne atrybuty, które go opisują. W ramach tego zadania poniżej zmiennej *hasCredit*, należy dodać zmienne, które pozwolą na zapisanie: nazwy uczelni, imienia studenta oraz jego numeru albumu (tylko cyfry), modyfikowanie wartości tych zmiennych nie powinno być możliwe.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacja tego zadania, nie wgrywaj całego projektu.

4. Wyrażenie warunkowe.

Student charakteryzuje się również aurą, która emanuje odpowiednim kolorem w zależności od jego poziomu zadowolenia. Poziom zadowolenia studenta jest wartością liczbową z zakresu od 0 do 20 i obliczą się ją zgodnie z poniższym wyrażeniem:

```
val satisfactionLevel = (Math.pow(Math.random(), (110 -  
studentPoints) / 100.0) * 20).toInt()
```

Zależności pomiędzy poziomem zadowolenia studenta a jego aurą zostały przedstawione w Tab. 1.

Wartość poziomu zadowolenia	Kolor aury
0-5	czerwony
5-10	pomarańczowy
11-15	purpurowy
16-20	zielony

Tab. 1

Wartość poziomu zadowolenia należy wyliczyć, używając wyrażenia podanego wcześniej, a następnie określić kolor aury, używając odpowiedniego wyrażenia warunkowego.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacja tego zadania, nie wgrywaj całego projektu.

5. Funkcje.

Z kodu który powstał do realizacji punktów 2-4 wydziel logikę odpowiedzialną za obliczanie aury studenta. Kod ten zamknij w funkcji *auraColor*, do funkcji przekaz parameter *studentPoints*, który domyślnie jest ustawiony na 2. Funkcja powinna być taka przygotowana aby była typu *Unit*.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacja tego zadania, nie wgrywaj całego projektu.

6. Nazwane argumenty funkcji.

Do kodu dodaj funkcję *printStudentStatus*, przyjmującą jako argumenty imię studenta oraz jego aurę. Funkcja ta powinna wyświetlać komunikat o statusie studenta w postaci "*\$name has a \$auraColor face color*", na przykład „Mike has a red face color”. Wywołaj funkcję *printStudentStatus* podając argumenty z odwrotną kolejnością niż są one zdefiniowane w nagłówku funkcji.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacja tego zadania, nie wgrywaj całego projektu.