

PROGRAMOWANIE MOBILNE

LAB6 – Programowanie w Swiftcie

1. Rozwiązania do poniższych zadań prześlij zbiorczo w jednym pliku pdf za pomocą platformy moodle <https://delta.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.

Do realizacji zadań skorzystaj ze środowiska Xcode, jeśli takowego nie posiadasz możesz użyć narzędzia dostępnego online <http://online.swiftplayground.run/> w celu weryfikacji poprawności składni i kompilacji kodu, użyj wersji ≥ 5.1 .

2. Deklarowanie zmiennych i interpolacja ciągu tekstowego.

Student może zdobyć zaliczenie z informatyki lub matematyki, dzięki czemu zbiera odpowiednią ilość punktów aby uzyskać wpis na kolejny semestr. Należy zdefiniować zmienną *points*, która pozwoli przechowywać informacje na temat ilości zdobytych przez studenta punktów. Skorzystaj z możliwości analizy tworzonych zmiennych i ustalenia typu tej zmiennej przez Swift i zadbaj o to aby zmienna była typu *Double* bez jawnego definiowania. Następnie za pomocą interpolacji ciągu tekstowego wyświetl informację na temat ilości punktów uzyskanych przez studenta np. „*Student has 5.0 points.*”.

Jako odpowiedź prześlij kod, który powstał na potrzeby realizacji tego zadania, nie wgrzywaj całego projektu.

3. 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 oblicz ją zgodnie z poniższym wyrażeniem:

```
var satisfactionLevel = round(pow(Double.random(in: 0...1), (110 - points) / 100) * 20)
```

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

Wartość poziomu zadowolenia	Kolor aury
0-5	red
6-10	orange
11-15	purple
16-20	green

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 realizacji tego zadania, nie wgrywaj całego projektu.

4. Klasy.

Każdy student ma imię, nazwisko oraz posiada punkty, zdefiniuj klasę *Student*, która to jednoznacznie opisze. Użyj metody inicjalizacyjnej do określenia wartości początkowych jej właściwości.

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

5. Enkapsulacja i metody klasowe.

Oznacz wszystkie właściwości zdefiniowane w klasie *Student* jako prywatne. Dodaj metodę *showStudent*, która wyświetli aurę studenta za pomocą wyrażenia warunkowego zdefiniowanego w punkcie 3, np. „Mike has a red face color” i będzie dostępna tylko w swoim module. Wywołaj metodę *showStudent* bez żadnych argumentów.

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

6. Dziedziczenie.

Imię jest charakterystyczne dla każdej osoby, nie tylko studenta. Zdefiniuj klasę bazową *Person*, w której umieścisz właściwości *name* i *lastname*. Klasa *Student* powinna być potomną klasy nadrzędnej *Person* i zawierać tylko właściwość *points*. Klasy *Person* i *Student* powinny mieć zdefiniowane metody inicjalizacyjne. Utwórz instancję klasy *Student* i za pomocą metody *showStudent* zaimplementowanej w ramach punktu 5 wyświetl aurę studenta.

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