

# Przykładowe kolokwium

## Informacje wstępne

1. Klonujemy repozytorium o nazwie PO-nr\_indeksu-zaliczenia do wybranego katalogu.
2. W PyCharm w sklonowanym katalogu z repozytorium tworzymy projekt o nazwie K1-nr\_indeksu, np. K1-12345.
3. W projekcie tworzymy 3 pliki: court.py, stadium.py oraz main.py
4. W trakcie kolokwium można korzystać z narzędzi wbudowanych w PyCharm lub zainstalowanych modułów takich jak: flake8 lub mypy w celu sprawdzenia typowania i PEP8.
5. W przypadku braku internetu student zapisuje projekt jako archiwum zip. Kolokwia będą zebrane np. za pomocą pendrive.
6. Rozwiązania po terminie: z uwagi na rozbieżność czasu na stanowiskach w pracowni bieżący czas będzie wyświetlony za pomocą rzutnika. Studenci zostaną uprzedzeni ustnie o zbliżającym się końcu czasu w okresie między 10 a 5 minut przed końcem kolokwium. Brak przesłania repozytorium w wyznaczonym czasie powoduje dodanie dodatkowych 5 minut, ale wtedy maksymalna ocena to 3,5. Powyżej 5 minut spóźnienia - kolokwium pozostaje bez sprawdzenia z oceną niedostateczną. Czas liczony jest od momentu udostępnienia poleceń do poprawnego wypchnięcia na zdalne repo na Githubie lub utworzenia pliku zip w wypadku braku internetu.

## Zadanie 1. Klasa Court (pol. Boisko) (18pt max.)

1. **(2pt)** Napisz klasę Court. Klasa powinna posiadać prywatne atrybuty instancyjne:
  1. width, (szerokość podawana w metrach), typ float
  2. length, (długość podawana w metrach), typ float
  3. address, (adres), typ str
  4. year\_built (rok budowy), typ int.
2. Zaimplementuj:
  1. **(5pt)** Inicjalizator z czterema argumentami, gdzie wartościami domyślnymi dla length będzie 150 metrów, a dla width 68 metrów. Zadbaj również o to aby inicjalizator respektował przepisy nt. wymiarów boisk sprzed 2008 roku. W przepisach tych długość mogła być z przedziału [90, 120] metrów, a szerokość z przedziału [45, 90] metrów. Jeśli przynajmniej jedna z wartości nie jest z odpowiedniego przedziału to dla obu pól powinny zostać ustawione wartości domyślne (W inicjalizatorze zachowaj kolejność argumentów jak w pt.1!).
  2. **(3pt)** Zaimplementuj właściwości (propercje) setter i getter dla każdego atrybutu.
  3. **(2pt)** Metodę area zwracającą powierzchnię boiska.

4. **(2pt)** Metodę statyczną `validate`, z jednym argumentem którym jest referencja do obiektu klasy `Court`, która sprawdzi czy podany rok budowy stadionu nie jest większy niż bieżący rok lub nie jest rokiem „ujemnym”. Jeśli rok jest niepoprawny to zmieni wartość atrybutu `year_built` na bieżący rok (moduł `datetime`).

3. Nadpisz:

1. **(2pt)** Odpowiednią metodę magiczną, która zwróci reprezentację tekstową bieżącej instancji klasy:

„Boisko wybudowane w roku 1999, o długości 100 metrów i szerokości 50 metrów.

Pole powierzchni: 5000 mkw.

Adres: Słoneczna 10, 10-100 Olsztyn.”

2. **(2pt)** Metody magiczne `__eq__`, `__ne__` porównujące obiekty po powierzchni boiska.

## Zadanie 2. Klasa `Stadium` (pol. `Stadion`) (13pt max.)

1. **(2pt)** Zaimplementuj klasę `Stadium` dziedziczącą po klasie `Court`. Klasa ta powinna posiadać trzy atrybuty instancyjne prywatne:
  1. `name` (nazwa stadionu np. PGE Narodowy), typu `str`.
  2. opcjonalny `common_name` (nazwa zwyczajowa, używana potocznie, nie każdy stadion ją ma) typu `str`.
  3. `capacity` (pojemność stadionu – możliwa ilość osób na trybunach), typu `int`, domyślnie 0.
2. **(4pt)** Nadpisz inicjalizator. Pamiętaj o tym, że atrybut `common_name` jest atrybutem opcjonalnym (pamiętaj o odpowiedniej kolejności argumentów!). Jeśli pojemność stadionu jest ujemna powinna być ustawiona domyślna wartość.
3. **(2pt)** Zaimplementuj właściwości (propercje) `setter` i `getter` dla każdego atrybutu. Jeśli dla `getterów` podane wartości argumentów nie spełniają założeń, wartość atrybutu nie powinna się zmieniać i powinien zostać wypisany odpowiedni komunikat.
4. **(2pt)** Nadpisz metody magiczne `__eq__`, `__ne__` porównujące obiekty biorąc pod uwagę jedynie powierzchnię boiska oraz pojemności stadionu .
5. **(2pt)** Nadpisz metodę magiczną, która zwróci reprezentację tekstową bieżącej instancji klasy:

a) „Boisko wybudowane w roku 1999, o długości 100 metrów i szerokości 50 metrów.

Pole powierzchni: 5000 mkw.

Adres: Słoneczna 10, 10-100 Olsztyn.

Nazwa: Słoneczny stadion.

Nazwa zwyczajowa: Słoneczko.

Pojemność stadionu: 10000 osób.”

lub gdy nie podano nazwy zwyczajowej:

b) „Boisko wybudowane w roku 1999, o długości 100 metrów i szerokości 50 metrów.

Pole powierzchni: 5000 mkw.

Adres: Słoneczna 10, 10-100 Olsztyn.

Nazwa: Słoneczny stadion.

Pojemność stadionu: 10000 osób.”

6. **(1 pt)** Pamiętaj również, żeby odziedziczone z klasy Court metody działały poprawnie w klasie Stadium.

## **Polecenia pomocnicze do testowania klas. Plik main.py z funkcją main() (Polecenia mają na celu pomoc w testowaniu napisanych klas)**

Część do klasy Court:

1. Stwórz obiekt o nazwie court\_1 i zainicjalizuj go domyślną szerokością i długością. Rok budowy i adres jak w punkcie 1.3.1.
2. Wypisz na ekran reprezentację obiektu court\_1.
3. Stwórz obiekt o nazwie court\_2 i zainicjalizuj go szerokością = 500 i długością = 500 oraz resztą danych z punktu 1.3.1.
4. Wypisz na ekran reprezentację obiektu court\_2.
5. Stwórz obiekt o nazwie court\_3 i zainicjalizuj go danymi z punktu 1.3.1.
6. Wypisz na ekran reprezentację obiektu court\_3.
7. Wypisz na ekran wartość atrybutu length obiektu court\_1.
8. Zmień wartość atrybutu year\_built obiektu court\_1 na 1990 i wypisz na ekran reprezentację obiektu court\_1 po zmianie stanu.
9. Wypisz na ekran powierzchnię boiska
10. Zmień wartość atrybutu year\_built obiektu court\_1 na 2030 i wypisz na ekran reprezentację obiektu court\_1 po zmianie stanu.
11. Użyj metody statycznej aby zwalidować rok budowy boiska court\_1
12. Wypisz na ekran reprezentację obiektu court\_1

Część do klasy Stadium:

1. (Stwórz obiekt o nazwie stadium\_1 i zainicjalizuj go domyślną szerokością i długością. Reszta argumentów jak w punkcie 2.4.1 a).
2. Wypisz na ekran reprezentację obiektu stadium\_1.
3. Stwórz obiekt o nazwie stadium\_2 i zainicjalizuj go danymi jak w punkcie 2.4.1 b).

4. Wypisz na ekran reprezentację obiektu stadium\_2
5. Zmień wartość atrybutu year\_built obiektu stadium\_1 na 2030 i wypisz na ekran reprezentację obiektu stadium\_1 po zmianie stanu
6. Użyj metody statycznej aby zwalidować rok budowy stadionu stadium\_1
7. Wypisz na ekran reprezentację obiektu stadium\_1
8. Sprawdź czy obiekty stadium\_1 i stadium\_2 są sobie równe
9. Zmień wartości atrybutów stadium\_1 na width=50, length=100
10. Sprawdź czy obiekty stadium\_1 i stadium\_2 są sobie równe
11. Sprawdź czy obiekty stadium\_1 i stadium\_2 nie są sobie równe
12. Zmień wartość atrybutu capacity obiektu stadium\_1 na 500
13. Sprawdź czy obiekty stadium\_1 i stadium\_2 są sobie równe

#### Punktacja

Zadanie 1: max: 18pt, Zadanie 1 zrobione poprawnie ale bez typowania i PEP8: max 16.2 pt

Zadanie 2: max: 13pt, Zadanie 2 zrobione poprawnie ale bez typowania i PEP8: max 11.7 pt

Ocena z kolokwium będzie obliczana wg poniższego wzoru:

- 0% - 50% punktów - ocena niedostateczna (2,0)
- 51% - 60% punktów - ocena dostateczna (3,0)
- 61% - 70% punktów - ocena dostateczna plus (3,5)
- 71% - 80% punktów - ocena dobra (4,0)
- 81% - 90% punktów - ocena dobra plus (4,5)
- 91% - 100% punktów - ocena bardzo dobra (5,0)
- przy czym każde zadanie musi być zaliczone na co najmniej 51% punktów aby uzyskać ocenę pozytywną, czyli Zadanie nr 1: minimum 9.18 pt, a Zadanie nr 2: minimum 6,63 pt.