

# Egzamin 30.01.2023, 10:30

## Uwagi wstępne:

1. Klonujemy repozytorium o nazwie `P0-nr_indeksu-zaliczenia` do wybranego katalogu.
2. W IntelliJ w sklonowanym katalogu z repozytorium tworzymy projekt o nazwie `E1-nr_indeksu`, np. `E1-123456`.
3. W przypadku braku internetu student zapisuje projekt jako archiwum zip. Kolokwia będą zebrane np. za pomocą pendrive.
4. Rozwiązania po terminie: z uwagi na rozbieżność czasu na stanowiskach w pracowni bieżący czas będzie podawany przez prowadzącego. 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 to, że 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.
5. Prowadzący mogą w trakcie kolokwium prosić o robienie commitów. Brak commita powoduje automatycznie ocenę niedostateczną.
6. Program, który się nie kompiluje, jest automatycznie niezaliczony.
7. **Jakakolwiek** próba **ściągnięcia** kończy egzamin z punktacją 0pt, bez możliwości poprawy (Zarządzenie Rektora Nr 5/2014).
8. **Jakakolwiek** próba **uwiecznienia egzaminu** kończy egzamin z punktacją 0pt, bez możliwości poprawy (Zarządzenie Rektora Nr 5/2014).
9. Jeśli chcesz skorzystać z **dokumentacji** poinformuj o tym prowadzącego.
10. Brak zachowania opisanej struktury katalogów powoduje brak sprawdzenia egzaminu.

## Zadanie 1 - interfejsy (8pt)

W paczce, którą właśnie ściągnąłeś znajduje się katalog `zadanie1` w którym znajdują się 4 pliki: `AcademicTeacher.java`, `Named.java`, `Professor.java` oraz `ProfessorDemo.java`. W swoim projekcie utwórz pakiet `zadanie1` i przerzuć do niego te pliki. Kod w danej postaci kompiluje się i działa bez błędów.

1. W pliku `ProfessorDemo.java` możesz znaleźć kod, który pomoże Ci w testowaniu Twoich zmian.
2. **(4pt)** dodaj kod tak by klasa `Professor` implementowała interfejs `Clonable` oraz by polecenie:  
`Named n = s;` w metodzie `main` było poprawne.

3. (4pt) napisz `Comparator`, który będzie odpowiedzialny za sortowanie profesorów alfabetycznie.<sup>1</sup>

## Zadanie 2 - Typy Generyczne (4pt)

Utwórz pakiet o nazwie `zadanie2`, tam umieść plik `Zadanie2.java`, a w nim napisz metodę statyczną, której deklaracja wykorzystuje **typy szablonowe** i która oblicza maksymalną wartość `ArrayList` zgodnie z naturalnym porządkiem elementów i kompiluje się bez błędów i ostrzeżeń. Przetestuj tę metodę.<sup>2</sup>

---

<sup>1</sup>W Zadaniu 1 do dopisania jest 10-20 linijek w zależności od formatowania kodu.

<sup>2</sup>Cały program ma koło 30 linijek kodu w zależności od formatowania.