

Egzamin 14.02.2023, 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 `E2-nr_indeksu`, np. `E2-123456`.
3. W przypadku braku internetu student zapisuje projekt jako archiwum zip. Egzaminy 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 (nie upload przez strone github)** 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. Brak zachowania opisanej struktury katalogów powoduje brak sprawdzenia egzaminu.
10. Jeśli chcesz skorzystać z **dokumentacji** poinformuj o tym prowadzącego.
11. Osoby piszące na własnych komputerach: brak nagrania całości egzaminu i całego ekranu powoduje ocenę niedostateczną bez sprawdzania kodu.
12. Osoby piszące tylko jedno zadanie mają 20 min mniej na egzamin.

Zadanie 1 - interfejsy (10pt)

W paczce, którą właśnie ściągnąłeś znajduje się katalog `zadanie1` w którym znajdują się 4 pliki: `Student.java`, `Named.java`, `Doktorant.java` oraz `DoktorantDemo.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 `DoktorantDemo.java` możesz umieścić kod, który pomoże Ci w testowaniu - nie będzie on oceniany.
2. (2pt) dodaj kod tak by klasa `Doktorant` implementowała interfejs `Clonable`.
3. (4pt) napisz `Comparator`, który będzie odpowiedzialny za sortowanie doktorantów alfabetycznie malejąco.
4. (4pt) zaimplementuj metodę `compareTo`, tak aby porównywała ona po długości łańcucha `name`. Pamiętaj o traktacie!

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 i zwraca minimalną i maksymalną wartość `ArrayList` zgodnie z **naturalnym porządkiem elementów** i kompiluje się bez błędów i ostrzeżeń. Przetestuj tę metodę dla trzech różnych `ArrayList`.