

**POJ**  
Laboratorium 8

**Zadania do samodzielnego wykonania :**

Zad 1:

Za pomocą programu w Javie stwórz katalog „exercise”. W nim stwórz trzy dowolne pliki i dwa katalogi.

Następnie stwórz obiekt InformationProvider, który będzie miał metodę displayInformation(File file) - ma ona wyświetlić następujące informacje dla każdego z plików / folderów:

- nazwa -
- ścieżka
- czy jest plikiem
- czy jest katalogiem
- czy plik istnieje

Zad 2:

Napisz program, który wczyta dane z pliku tekstowego i zsumuje co drugą linię pliku. Plik znajduje się na dostępnym w zakładce ‘Pliki’ („Input”).

Zad 3:

Napisz program, który przeanalizuje katalogi w basedir (dostępne w zakładce ‘Pliki’, „basedir.zip”). Program ma wykazać który z katalogów ma najwięcej plików.

Zad 4:

Stwórz katalogi inbound i outbound i przykładowe pliki w katalogu inbound (poza programem).

Napisz program który ma przetworzyć wszystkie pliki w katalogu inbound w następujący sposób:

- ma zamienić wszystkie spacje w treści plików (nie nazwie!) na podkreślenie “\_”
- ma każdą literę “przesunąć” o trzy pozycje w alfabecie (a->d, b->e) itp. Uwzględnić duże litery.
- plik przenieść do katalogu outbound i dodać do jego nazwy obecną datę i rozszerzenie „.out”
- plik ma być usunięty z katalogu inbound

Zad 5:

Napisz program, który poda następne 100 roczników przestępnych za pomocą klasy LocalDate.

Zad 6:

Napisz program, w którym użytkownik poda godzinę (z minutami) w Polsce i wybierze kraj / miasto, a program pokaże mu przeliczoną **datę i godzinę** w tym kraju / mieście. Przygotuj 3 takie kraje do wyboru z różnych rejonów świata. Przyjmij, że użytkownik chce datę dzisiejszą. Jeśli przesunięcie czasu sprawi, że będzie nowy dzień, uwzględnij to.

Zad 7:

Przygotuj interfejs Figura który określa dwie metody: „obliczPole” i „obliczObwód”. Następnie przygotuj trzy różne klasy które implementują ten interfejs na różne sposoby. Pokaż w pętli jak działają.

Zad 8:

Stwórz klasę Student - niech ma imię, nazwisko i numer indeksu. Niech implementuje interfejs Comparable. Stwórz tablice takich Studentów i posortuj ich za pomocą Arrays.sort(). Studenci mają być posortowani według indeksu. (indeks ma mieć format PJATK, np. s12345)

Rozwiązanie zadań powinno być udostępnione w na indywidualnym repozytorium github studenta maksymalnie do końca czwartego dnia po zajęciach:

- zajęcia w poniedziałek -> termin do piątku 23:59:59
- zajęcia w środę -> termin do niedzieli 23:59:59
- zajęcia w czwartek -> termin do poniedziałku 23:59:59

Zadania powinny być zrealizowane w osobnych klasach w ramach jednego projektu, a umieszczony w prywatnym repozytorium, które jest udostępnione prowadzącemu (<https://github.com/plucins>)