

## Wariant 721

- Stwórz nowy projekt a w nim pakiet `pl.edu.uwm.airplanes` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Airplane` z prywatnym polem `maxAltitude` (typu `double`, pol. maksymalna wysokość).
- Zaimplementuj konstruktor parametryczny ustawiający pole klasy z argumentu.
- Nadpisz metodę `equals` i `hashCode()`, tak aby porównywała obiekty według pola w klasie.
- Stwórz klasę `TestAirplane` z metodą `main`. W metodzie `main` sprawdź działanie metod `equals` i `hashCode()`.

Zadanie należy umieścić na pendrive. Cały projekt ma być spakowany jako archiwum zip. Nazwa według schematu `NUMERWARIANTU_NUMERALBUMU.zip` np. `602_123456.zip`.

## Wariant 722

- Stwórz nowy projekt a w nim pakiet `pl.edu.uwm.ships` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Ship` z prywatnym polem `tonnage` (typu `double`, pol. tonaż).
- Zaimplementuj konstruktor parametryczny ustawiający pole klasy z argumentu.
- Nadpisz metodę `equals` i `hashCode()`, tak aby porównywała obiekty według pola w klasie.
- Stwórz klasę `TestShip` z metodą `main`. W metodzie `main` sprawdź działanie metod `equals` i `hashCode()`.

Zadanie należy umieścić na pendrive. Cały projekt ma być spakowany jako archiwum zip. Nazwa według schematu `NUMERWARIANTU_NUMERALBUMU.zip` np. `602_123456.zip`.

## Wariant 723

- Stwórz nowy projekt a w nim pakiet `pl.edu.uwm.bicycles` i umieść w nim oddzielne pliki dla poniższych klas.
- Stwórz klasę `Bicycle` z prywatnym polem `frameSize` (typu `double`, pol. rozmiar ramy).
- Zaimplementuj konstruktor parametryczny ustawiający pole klasy z argumentu.
- Nadpisz metodę `equals` i `hashCode()`, tak aby porównywała obiekty według pola w klasie.
- Stwórz klasę `TestBicycle` z metodą `main`. W metodzie `main` sprawdź działanie metod `equals` i `hashCode()`.

Zadanie należy umieścić na pendrive. Cały projekt ma być spakowany jako archiwum zip. Nazwa według schematu `NUMERWARIANTU_NUMERALBUMU.zip` np. `602_123456.zip`.