1.A

Klasy abstrakcyjne używane są gdy dziedziczą po nich klasy mające ze sobą wspólne nie tylko metody, ale również pola klas, których zastosowanie w klasie abstrakcyjnej pozwoli ograniczyć powtarzanie na potencjalnie wielu klasach tego samego kodu.

Interfejsy używane są w sytuacji gdy mają opisywać one zachowanie klas poprzez definiowanie pustych metod, które muszą być nadpisane w klasach implementujących interfejs, lub poprzez metody domyślne zawierające stałą funkcjonalność dla wszystkich klas.

Warto również zaznaczyć iż klasy mogą implementować wiele interfejsów, ale mogą rozszerzać tylko jedną klasę abstrakcyjną.

1.B

Bardzo dużą przewagą listy nad zwykłą tablicą jest możliwość powiększania się. W przypadku tablic po utworzeniu i zadeklarowaniu danej ilości elementów nie będzie możliwa zmiana jej wielkości. Listy umożliwiają również bardzo łatwe dodawanie i usuwanie elementów do jej środka. Przesuwają one automatycznie wszystkie elementy w taki sposób, aby zachowana była ich ciągłość. Aby osiągnąć taki sam efekt w przypadku tablic musielibyśmy samodzielnie stworzyć metody dla dodawania i usuwania elementów.

Różnią się one również sposobem tworzenia.

W taki sposób tworzymy Listy, oraz dodajemy do nich elementy przy pomocy metody add().

```
List<Integer> testList = new ArrayList<>();
testList.add(1);
testList.add(2);
testList.add(3);
```

Natomiast tablice możemy utworzyć na 2 sposoby:

a) Przypisując wartości do tablicy przy jej tworzeniu jak w przypadku testArray,

b) Tworząc tablicę przy użyciu new int[3] podając w nawiasach kwadratowych wielkość tablicy, a następnie dodanie wartości do odpowiednich indeksów w tablicy (testArray2).

```
int[] testArray = {1, 2, 3};
int[] testArray2 = new int[3];
testArray2[0] = 1;
testArray2[1] = 2;
testArray2[2] = 3;
```

2.

Szacowana złożoność obliczeniowa – O (log n)

Szacowana złożoność pamięciowa - O (1)