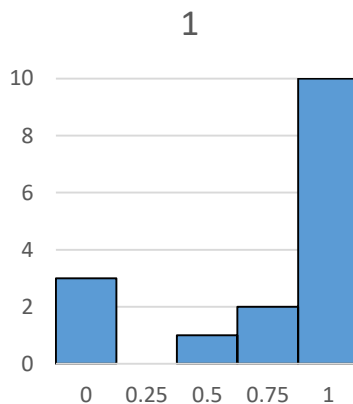


Komentář k řešení

1) Co bude výsledkem následujícího kódu?

```
for(int i = 0; i < 10; i = i++) {i = i++;}  
System.out.println(i);
```

V kódu je několik problémů, které zabraňují vykonání druhé řádky. Primárně je problém ve volání lokálně definované proměnné `i` mimo její scope, což vede na compile error. Cyklus sám o sobě je nekonečný kvůli `i = i++` v definici for cyklu.



2a) Napište pseudokódem metodu pro vyřešení kvadratické rovnice. (včetně definice hlavičky metody, deklarace proměnných, předání výsledku)

```
??? solveQuadraticEq(???){  
    ...  
    return ???;  
}
```

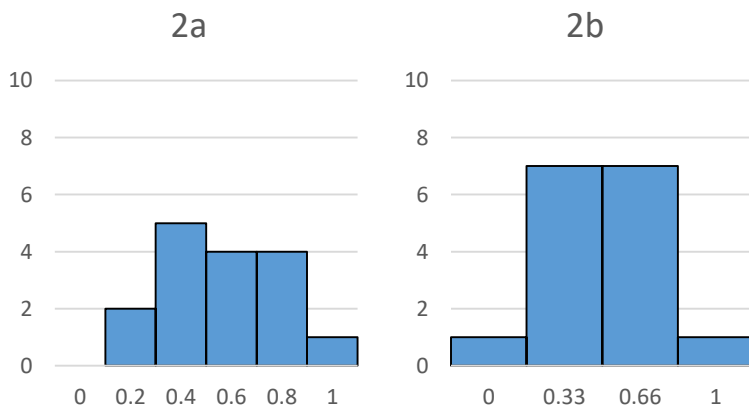
Zde mi primárně zájímalo, jestli nějakým způsobem řešíte:

- dělení nulou v případě, že $a=0$,
- smysluplná struktura a datové typy vstupů
- smysluplné vrácení výsledku
- vnitřní logika metody
- diskriminant < 0

2b) Uveďte předpoklady, omezení a přesnost předchozího řešení.

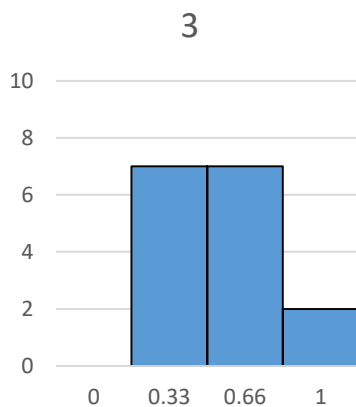
O metodě bych čekal dokumentaci, která zahrne například:

- předpoklady – na vstupu je kvadratická rovnice, tedy $a \neq 0$,
- omezení – řešení pouze na oboru reálných čísel
- přesnost – dle použitého datového typu float/double...



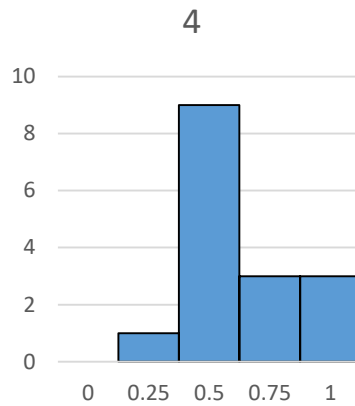
3) Na jaké problémy mohu narazit při konverzi String -> int?

Na první pohled je patrný problém v situaci, kdy v řetězci není text, který by bylo možné interpretovat jako číslo. Nicméně další zajímavé problémy mohou být například při pokusu o parsování příliš velkého čísla (existuje max_value pro datový typ int dle platformy) nebo při nutnosti zpracovat lokalizovaně formátované číslo (oddělovač tisíců).



4) Vysvětlete pojmy primární klíč, cizí klíč, kompozitní klíč, kandidátní (potenciální) klíč.

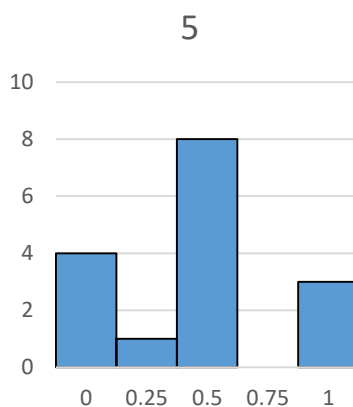
Primární a cizí klíč nebyl problém. Kompozitní klíč je takový, který využívá kombinaci několika sloupců. Kandidátním klíčem se rozumí atribut, který svými vlastnostmi vyhovuje podmínkám na primární klíč.



5) Jaké rozměry bude mít tabulka výsledku následujícího dotazu? (zdůvodněte odhad)

```
-- tabulka obce je číselník obcí v ČR
-- tabulka obyvatel obsahuje všechny fyzické osoby s trvalým pobytem v ČR
SELECT *
FROM obce, obyvatel
WHERE obce.name = "Plzeň" ;
```

Trik byl v rozpoznání kartézského součinu kvůli nedefinování propojení tabulek **obce** a **obyvatel** v dotazu. Kvůli WHERE klauzuli je redukována dimenze tabulky **obce** na 1 x n (řádky x sloupce). Tabulka **obyvatel** má rozměr přibližně 10 000 000 x m. Odhad rozměru výsledku je tedy 10 000 000 x (n+m), kde n a m jsou počty atributů daných entit.

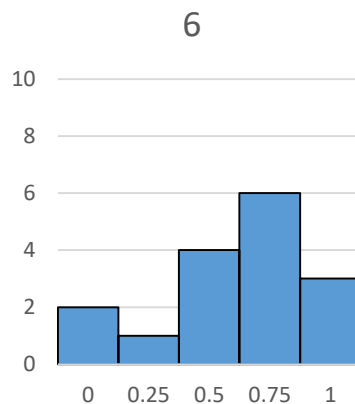


6) Zkouškový termín úspěšně absolvovalo 6 studentů se známkami {1, 1, 1, 1, 2, 3}, 3 studenti v termínu neuspěli, 1 student měl omluvenou neúčast. Určete stř. hodnotu, modus, medián známek. Jaká byla úspěšnost termínu?

Zde bylo k zamyšlení, co udělat se studenty, kteří neuspěli a se studentem, který byl omluven. Největší smysl by zde dávalo pro neúspěšné uvažovat hodnotu 4 a omluveného ve statistikách dále nezahrnovat. Volba hodnoty 0 pro omluveného by sice dávala určitý sémantický smysl, ale dělala by jednosměrný bias.

Modusem se rozumí nejčastější hodnota, v tomto případě hodnota 1 (4x v datasetu).

Úspěšnost termínu je běžně počítána jako úspěšní/(úspěšní+neúspěšní), tedy 6/9.



7) V jaké situaci byste použili klouzavý průměr? Uveďte jeho vzorec.

Typicky se s touto technikou setkáte u časových řad nebo realtime senzorů a slouží k jednoduchému „vyhlazení“ ostrých výchylek v jednom bodě. Hodnota v určitém bodě je tvořena průměrem z n předcházejících bodů, tedy vzorcem např. $f(x) = 1/n * \text{sum}(\text{raw}[x], \text{raw}[x-1], \dots, \text{raw}[x-n+1])$, kde raw je vektor čistých vstupních dat.

