

## 1. Sneakers

Дадена е класа **Sneakers** во која се чуваат информации за патики и тоа: модел (текстуална низа), големина (цел број) и цена (децимален број). Во класата се дефинирани сите потребни конструктори и деструктор, оператори, set и get методи. **Оваа класа не треба да се менува.**

Да се дефинира класа **SneakersShop** во која се чуваат податоци за: **(5 поени)**

- име на продавницата за патики (низа од најмногу 50 знаци)
- патики што ги нуди продавницата (динамички алоцирана низа од објекти од класата Sneakers)
- вкупен број на објекти од класата Sneakers (цел број)
- Поскапување (изразено во проценти, цел број) - ист за сите производи достапни во продавницата и иницијално има вредност 5. Оваа вредност може да се менува.

За класата да се имплементираат:

- потребните конструктори и деструктор **(5 поени)**
- да се преоптовари операторот **+=** за додавање нов објект од класата Sneakers во продавницата. **(5 поени)**
- метод **float checkItemAvailability(char\* model, int size)** - што ќе провери дали моделот што се проследува како прв влезен аргумент е достапен во продавницата за патики во големината што се проследува како втор влезен аргумент. Ако моделот е достапен во дадената големина (на пример, NIKE FLEX RUNNER, големина 38), методот ја враќа цената на патиките, во спротивно враќа вредност 0. **(5 поени)**
- Оператор **++** (во префикс нотација) за зголемување на цената на сите патики за вредноста на покачувањето. **(5 поени)**

Да се имплементира класа **OnlineSneakersShop** во која покрај основните податоци за продавницата, дополнително се чува и: **(5 поени)**

- Url на веб страницата на онлајн продавницата за патики (низа од 100 знаци).
- Попуст (изразен во проценти, цел број) - ист за сите производи достапни во онлајн продавницата и иницијално има вредност 10. Тој не може да се менува.

За класата да се имплементираат потребните конструктори и деструктор. **(5 поени)** Дополнително, да се обезбеди и следново:

- метод **float checkItemAvailability(char\* model, int size)** - што ќе провери дали моделот што се проследува како прв влезен аргумент е достапен во онлајн продавницата за патики во големината што се проследува како втор влезен

аргумент. Ако моделот е достапен во дадената големина (на пример, NIKE FLEX RUNNER, големина 38), методот ја враќа цената на патиките намалена за попустот достапен во онлајн продавницата. Во спротивно, методот враќа вредност 0. **(5 поени)**

За двете класи да се дефинира оператор за печатење <<. Форматот на печатење на двете групи погледнете го во првите два [тест](#) примера. **(5 поени)**

**For example:**

| Input  | Result   |
|--|--|
| 1<br>TEST_SHOP   | TESTING CLASS SneakersShop: CONSTRUCTOR AND OPERATOR <<<br>Sneakers shop: TEST_SHOP<br>List of sneakers:<br>EMPTY  |
| 4<br>BuzzSneakers<br>https://www.buzzsneakers.com/<br>4<br>NIKE_AIR_JORDAN_LEGACY_312 40 9590<br>NIKE_AIR_MAX_96_II_PREMIUM 41 12590<br>CONVERSE_CHUCK_70 39 4232<br>NEW_BALANCE_327 40 6590   | TESTING CLASS OnlineSneakersShop, CONSTRUCTOR, += and <<<br>Online sneakers shop: BuzzSneakers URL: https://www.buzzsneakers.com/<br>List of sneakers:<br>EMPTY<br>Online sneakers shop: BuzzSneakers URL: https://www.buzzsneakers.com/<br>List of sneakers:<br>1. Model: NIKE_AIR_JORDAN_LEGACY_312 Size: 40 Price: 9590<br>2. Model: NIKE_AIR_MAX_96_II_PREMIUM Size: 41 Price: 12590<br>3. Model: CONVERSE_CHUCK_70 Size: 39 Price: 4232<br>4. Model: NEW_BALANCE_327 Size: 40 Price: 6590 |
| 5<br>1<br>BuzzSneakers<br>https://www.buzzsneakers.com/<br>6<br>NIKE_AIR_JORDAN_LEGACY_312 40 9590<br>NIKE_AIR_MAX_96_II_PREMIUM 41 12590<br>CONVERSE_CHUCK_70 39 4232<br>NEW_BALANCE_327 40 6590<br>NIKE_AIR_MAX_96_II_PREMIUM 42 12590<br>NIKE_AIR_MAX_96_II_PREMIUM 43 12590<br>NIKE_AIR_MAX_96_II_PREMIUM 42 | TESTING METHOD checkItemAvailability<br>Price for model NIKE_AIR_MAX_96_II_PREMIUM with size: 42 is: 11331   |

## 2. Evaluator

Да се дефинира апстрактна класа `Evaluator` за евалуација на алгоритми за класификација на елементи во две категории (позитивна (+) или негативна (-)). Класата да има два чисто виртуелни методи:

- `double evaluate ()`
- `void print()`

Во класата дополнително да се чуваат две низи од најмногу 100 знаци за:

- Вистинските категории на елементите
- Предвидените категории на елементите

Низите мора да имаат иста големина. Дозволените вредности во двете NULL-терминирани низи се само знаците + (означува позитивна категорија) и - (означува негативна категорија). **(5 поени)**

Појасување на текстуалните низи: На пример, ако двете NULL-терминирани низи се "++++"(за вистинските категории) и "+---"(за предвидените категории) тоа означува дека првиот и четвртиот елемент се од позитивна категорија и се точно предвидени како позитивна категорија. Вториот и третиот елемент се од позитивна категорија, но се грешно предвидени како елементи од негативна категорија.

Од класата `Evaluator` да се изведат класите `AccuracyEvaluator` и `SensitivityEvaluator`. **(5 поени)** Во класите да се препокрие методот `evaluate()` на следниот начин:

- Во `AccuracyEvaluator`, точноста се пресметува како количник на бројот на точни предвидувања и бројот на елементи. Пример, за вистинските категории `++++` и предвидените категории `--++`, точноста е 0.5 (последните два елемента се точно предвидени како +, а првите два елемента се грешно предвидени како -, па  $2/4 = 0.5$ ). **(10 поени)**
- Во `SensitivityEvaluator`, сензитивноста се пресметува како количник на бројот на точно предвидени елементи од позитивна категорија со бројот на елементи чија што вистинска категорија е позитивна. Пример за вистинските категории `+++-` и за предвидените категории `+---`, точно предвиден во позитивна категорија е само 1 елемент (првиот елемент), вкупно елементи со вистинска позитивна категорија се 3, па сензитивноста е  $1/3 = 0.33$ . **(10 поени)**

Методот `print()` печати информации за евалуаторите и тоа во следниот формат:

- За `AccuracyEvaluator`: `Number of data points:` [бројот на елементи што се предвидуваат] `Prediction Accuracy:` [пресметаната точност]
- За `SensitivityEvaluator`: `Number of data points:` бројот на елементи што се предвидуваат] `Prediction Sensitivity:` [пресметаната сензитивност] **(5 поени)**

За класите да се дефинираат потребните методи за правилно функционирање на програмата. Со помош на исклучоци да се спречи креирање на објекти од изведените класи на класата `Evaluator`, ако двете низи од знаци имаат различни големина. Во такво сценарио да се фрли исклучок од тип `WrongLabelsFormatException`. **(5 поени)** Со исклучокот потребно е да се справите во функцијата `main`. Кога ќе се фати исклучокот, да се испечати соодветна порака како во [тест](#) примерите. **(5 поени)** Сите низи во [тест](#) примерите ќе бидат составени исклучиво од знаците `+` и `-`.

Да се дефинира оператор `>=` за споредба на два евалуатора според резултатот од методот `evaluate`. **(3 поени)**

Да се дефинира глобална функција `Evaluator * highestScore (Evaluator ** evaluators, int n)`, што ќе врати покажувач кон евалуаторот со најголем резултат од методот `evaluate`. Ако има повеќе евалуатори со максимален резултат од методот `evaluate`, да се врати последниот. **(7 поени)**

**For example:**

| Input                  | Result   |
|------------------------|--|
| 1<br>++++<br>--++      | TESTING CREATION OF ACCURACY EVALUATOR. EXCEPTION CAN HAPPEN HERE!<br>0.5<br>Number of data points: 4 Prediction Accuracy: 0.5   |
| 2<br>+++--<br>+---+    | TESTING CREATION OF ACCURACY EVALUATOR. EXCEPTION CAN HAPPEN HERE!<br>0.333333<br>Number of data points: 4 Prediction Sensitivity: 0.333333                                  |
| 3<br>+++-----<br>----- | TESTING CREATION OF BOTH EVALUATORS. EXCEPTION CAN HAPPEN HERE!<br>Number of data points: 10 Prediction Accuracy: 0.7<br>Number of data points: 10 Prediction Sensitivity: 0 |
| 1<br>++++<br>-+-       | TESTING CREATION OF ACCURACY EVALUATOR. EXCEPTION CAN HAPPEN HERE!<br>Labels must have the same length   |