

Да се създаде сървър, който служи като помощно средство на програмисти при работа с регулярни изрази. Сървърът позволява тестване на създаден от потребителя регулярен израз и съхранява вече създадени такива във файл.

Създайте клас `Regex` със следните полета: [5.]

- `id` – цяло число, идентификатор на регулярния израз
- `pattern` – символен низ, шаблона на израза
- `description` – символен низ, описва с ключови думи за какво служи дадения регулярен израз (например `email`)
- `rating` – цяло число, потребителска оценка за качеството на израза

`Regex` да съдържа и статично поле `int nextId` с първоначална стойност 0.

Създайте конструктор, който инициализира полетата `pattern` и `description` чрез параметри, полето `rating` – с 0 и полето `id` със следващата целочислена стойност на `nextId`. [5.]

Сървърът трябва да може да запазва гореописаните регулярни изрази. Създайте функционалности за четене и запис на списък от `Regex` във файл. Да не се допуска добавяне на `Regex`, чийто `pattern` вече съществува във файла. [10т.]

Създайте клас `RegexTester` със статичен метод `List<Boolean> test(Regex regex, String[] strings)`. Методът да проверява низовете от масива `strings` спрямо подадения регулярен израз. Ако низът отговаря на шаблона, резултатът е `true`. Методът връща списък с булеви стойности – резултатите от всички проверки. [10т.]

При свързване към сървъра клиентите имат 2 опции за работа – създаване на собствен регулярен израз или търсене на такъв във файла.

При създаване на собствен `Regex` клиентът попълва полетата за шаблон и описание. След това въвежда низове за тестване разделени с нов ред. Сървърът провежда тестването на регулярния израз чрез `RegexTester` и изпраща резултата на клиента. Клиентът избира, дали иска да добави своя регулярен израз към файла. [10т.]

Ако клиентът избере опция за търсене на съществуващ израз, той въвежда ключова дума. Сървърът филтрира списъка от файла и връща на клиента списък с `Regex`, чиито описания съдържат ключовата дума. Списъкът е подреден по полето `rating` в низходящ ред. Клиентът избира `Regex` от списъка като въведе неговото `id`. След това въвежда низове за тестване. Тестването се провежда по аналогичен начин. Клиентът има опцията да промени `rating` на регулярния израз с плюс или минус едно. [10т.]

Сървърът трябва да може да обслужва няколко клиентски програми едновременно. [10т.]

При работа със споделени ресурси да се осигури необходимата синхронизация. [10т.]

НЕ пишете `get/set` методи. Приемете, че те вече са написани и може да ги използвате.

НЕ пишете код на клиентската програма. Реализирайте единствено сървъра.