

Лабораторна робота №7

Робота з файлами та серіалізація. Робота з текстом та регулярними виразами

Мета

Вивчити можливості роботи з файлами. Освоїти практично переваги використання серіалізації.

Вивчити можливості роботи з рядками. Закріпити оброблення параметрів командного рядка. Освоїти використання класів `StreamReader` та `StreamWriter`. Навчитися працювати з регулярними виразами.

Завдання 1

Проаналізуйте інформацію, яку можна отримати за допомогою API (див. Додаток А).

Реалізувати тип даних (клас), що відповідає сутностям (об'єктам реального світу), про які надається інформація через API. Може знадобитися створення кількох класів. Клас (класи) повинен включати поля/властивості для зберігання всієї інформації, що надається.

Розробити консольну програму, що дозволяє десеріалізувати інформацію в колекцію об'єктів створеного типу даних.

Реалізувати такі функції:

- додавання нового запису до колекції;
- видалення запису з колекції;
- перегляд усіх записів;
- зміна будь-якого запису;
- пошук записів по кожному з полів;
- експорт будь-якого запису до текстового файлу.

Програма має зберігати всі введені/змінені дані під час виходу з програми, якщо обрано пункт меню «Вихід». При запуску програма має перевіряти наявність файлу з даними та завантажувати їх автоматично, якщо файл, створений у попередньому сеансі роботи, доступний та коректний. Збереження даних має бути реалізоване з використанням механізму серіалізації, завантаження – з використанням механізму десеріалізації.

Також у програмі необхідно передбачити оброблення можливих винятків (наприклад, файл з даними відсутній, пошкоджений, введені некоректні дані користувачем тощо).

Завдання 2

Розробити консольну програму, призначену для перевірки рядків на відповідність заданому шаблону. Результатом роботи програми має бути виведення інформації про те, які рядки відповідають заданому шаблону, а які – ні.

Якщо користувач вказав ключ `/s`, програма має перевірити всі слова, передані після ключа в командному рядку.

Якщо користувач вказав ключ /f та ім'я файлу, програма має перевірити всі слова у вказаному файлі.

Якщо користувач вказав ключ /? або /h, програма має вивести довідкову інформацію про її призначення та опис усіх ключів.

Якщо користувач запустив програму без ключів, вона має вимагати рядки у користувача та виводити результати перевірки; введення рядків завершується, якщо користувач ввів рядок "end".

Шаблон повинен бути поставлений за допомогою регулярного виразу.

Варіанти індивідуальних завдань наведено у додатку Б.

Кожен елемент класу повинен містити документуючий коментар (///), що описує його призначення.

Вимоги до звіту

Звіт з даної лабораторної роботи повинен включати:

1. титульний лист із зазначенням номера та назви лабораторної роботи;
2. мета роботи;
3. діаграму класів;
4. вихідні тексти розроблених програм;
5. результати роботи розроблених програм (скріншоти або тексти, що виводяться програмами);
6. тестові набори (обов'язково включити тестові набори, що призводять до появи винятків);
7. висновки, у яких необхідно відобразити призначення всіх розроблених класів.

Контрольні питання

1. Які класи .NET Framework призначені для роботи з файлами?
2. Що таке серіалізація?
3. Що таке десеріалізація?
4. Що таке маршалінг/демаршалінг?
5. Які види серіалізації існують? У чому полягають їхні переваги/недоліки?
6. Що таке виняток?
7. Для чого призначене оброблення винятків?
8. Як можна обробити винятки всіх типів?
9. Як можна обробити лише певний тип винятків?
10. Як отримати доступ до параметрів командного рядка в консольній програмі?
11. Для чого призначений клас StreamReader?
12. Для чого призначений клас StreamWriter?
13. Які конструктори має клас String?
14. Які методи є у класу String? Наведіть приклади.
15. Що означає факт, що String є незмінюваним типом даних?
16. Які виняткові ситуації можливі під час роботи з файлами?
17. Які існують режими відкриття файлів?

- 18.Що таке регулярні вирази? Для чого вони призначені?
- 19.Для чого призначений клас Group?
- 20.Для чого призначений клас Capture?
- 21.Для чого призначений клас Match?
- 22.Для чого призначений клас RegEx?

Додаток А. Варіанти завдань

<i>№</i>	<i>Дані</i>	<i>Формат</i>	<i>Посилання</i>
1.	Курс валют НБУ на поточну дату	JSON	https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json
2.	Курс валют НБУ на поточну дату	XML	https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?xml
3.	Інформація про країни	XML	http://api.worldbank.org/v2/country
4.	Курс валют ЄЦБ на поточну дату	XML	https://www.ecb.europa.eu/stats/eurofxref/eurofxref-daily.xml
5.	Курс валют Монобанк на поточну дату	JSON	https://api.monobank.ua/bank/currency
6.	Інформація про транспортні засоби, що знаходяться в розшуку	JSON	https://data.gov.ua/dataset/9b0e87e0-eea3-4f14-9547-03d61b70abb6/resource/e43a82da-89e1-4bbb-820c-bd04ab7a0c89/download/car-swanted.json
7.	Реєстр державних послуг	JSON	https://guide.dii.gov.ua/register/download/json/
8.	Український індекс міжбанківських ставок	JSON	https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange?json
9.	Український індекс міжбанківських ставок	XML	https://bank.gov.ua/NBUStatService/v1/statdirectory/exchange
10.	Прогноз погоди по Харкові	JSON	https://api.open-meteo.com/v1/forecast?latitude=49.98&longitude=36.25&hourly=temperature_2m
11.	Дані про декларативні документи з Реєстру будівельної діяльності	JSON	https://data.gov.ua/dataset/d1e27dd5-a9b4-4b20-b5f7-a7b3163e9e0e/resource/ead61051-b018-40df-a86b-16892c63a8a6/download/export_data_gov_ua_data_2022-02-22.zip

12.	Список виступаючих на конференції	JSON	https://gist.github.com/planetoftheweb/2c2f3b03b72a7f2ae923
13.	Підрозділи поліції	JSON	https://data.police.uk/api/forces
14.	Інформація про людину	JSON	https://randomuser.me/api
15.	Час сходу та заходу сонця	JSON	https://api.sunrise-sunset.org/json?lat=49.9935&lng=36.2304
16.	Інформація про країни	JSON	http://api.worldbank.org/v2/country?format=json
17.	Курси валют національного банку Чехії	XML	https://www.cnb.cz/cs/financi_trhy/devizovy_trh/kurzy_devizoveho_trhu/denni_kurz.xml
18.	Інформація про Україну	JSON	http://api.worldbank.org/v2/country/ua?format=json
19.	Університети світу	JSON	https://raw.githubusercontent.com/Hipo/university-domains-list/master/world_universities_and_domains.json
20.	Виробники авто	JSON	https://vpic.nhtsa.dot.gov/api/vehicles/getallmakes?format=json

Додаток Б. Варіанти для завдання 2

<i>Варіант</i>	<i>Завдання</i>
1.	Рядки, що містять як слово ціле число. Приклад рядків, які підходять: Year is 2022., 1 is a number, 3.1415 matches because . is not a word char», «Lab 7 is very simple». Приклад рядків, які не підходять: Not2Bad, No digits here, 2Good4You, COVID19.
2.	Рядки, що містять коректні ідентифікаційні номери транспортних засобів (VIN-коди). Приклади рядків, які підходять: «WP1ZZZ92ZBLA03352», «WBSGZ01010L587108», «Y6DTF698KE0332004», «ХТЕ110270R0282981». Приклади рядків, які не підходять: «VINVIN123456», «12345678987654321».
3.	Рядки, що містять коректну суму в доларах із можливими роздільниками тисяч. Приклади рядків, які підходять: \$5.72, \$1,000,000. Приклади рядків, які не підходять: \$5.721, \$5,42,16,5 dollars.
4.	Рядки, що містять коректні номерні знаки транспортних засобів України типу 1 згідно з ДСТУ 4278. Приклади рядків, що підходять: «AX0200BA», «KX7777PP», "BI5419EC". Приклади рядків, які не підходять: «AXAXAX», «TT0901TT», «EC1234AA».
5.	Рядки, що містять коректні номери телефонів мобільних операторів України. Приклади рядків, які підходять: «(050) 310-20-89», «0971125678», «073-123-45-76». Приклади рядків, які не підходять: «0112345678», «(077) 212-45-08», «09548567нуль».
6.	Рядки, що містять двійковий запис числа, кратного 3. Приклад рядків, які підходять: «0», «10010». Приклад рядків, які не підходять: «00101», «Not a number», «1 1», «0 0».

7.	<p>Рядки, що є смайликами:</p> <ul style="list-style-type: none">– першим символом є або ; або: один раз;– далі може йти символ – скільки завгодно разів (у тому числі нуль разів);– наприкінці обов'язково йде деяка кількість (не менше однієї) однакових дужок із наступного набору: (,), [,];– усередині смайлика не може зустрічатися жодних інших символів. <p>Приклад рядків, які підходять: «:»», «;-----[[[[[[]]»].</p> <p>Приклад рядків, які не підходять: ":-)]", ";--".</p>
8.	<p>Рядки, що містять слово всередині довільного тексту, що не містить дужок, у дужках. Приклад рядків, які підходять: "good (excellent) phrase", "good (too bad) phrase", "good ((recursive)) phrase". Приклад рядків, які не підходять: "word () is not () in brackets", "bad((() recursive) phrase", "no brackets here".</p>
9.	<p>Рядки, що містять «cat» як слово. Приклад рядків, які підходять: cat, catapult and cat, catapult and cat and concatenate. Приклад рядків, які не підходять: catcat, concat, Cat.</p>
10.	<p>Рядки, що не містять розділювачів на початку та в кінці. Приклад рядків, які підходять: «Good string», «». Приклад рядків, які не підходять: « bad string», «bad string «, « very bad string «.</p>
11.	<p>Рядки, що містять правильні MAC-адреси. Приклад рядків, які підходять: «01:32:54:67:89:AB», «aE:dC:cA:56:76:54». Приклад рядків, які не підходять: «01:33:47:65:89:ab:cd», «01:23:45:67:89:Az».</p>
12.	<p>Рядки, що відповідають шаблону «YYYY/MM/DD», і містять коректну дату в діапазоні від 1990 до 2025 року. Приклад рядків, які підходять: «2012/05/02», «1990/11/05». Приклади рядків, які не підходять: «2029/03/02», «2022/25/25».</p>

13.	Рядки, що містять коректні IPv4 адреси в десятковому поданні з точками. Приклади рядків, які підходять: «195.88.72.125», «127.0.0.1». Приклад рядка, який не підходить: «256.256.256.256».
14.	Рядки, що містять доменні імена для протоколів http і https, з обов'язковим слешем наприкінці. Приклади рядків, які підходять: «http://khai.edu», «https://amazon.com/». Приклади рядків, які не підходять: «khai.edu», «http://khai..edu».
15.	Рядки, що містять правильні hex ідентифікатори кольору HTML. Приклади рядків, які підходять: «#FFFFFF», «#FF3421», «#00ff00». Приклади рядків, які не підходять: «232323», «f#fddee», «#fd2».
16.	Рядки, що містять коректні номерні знаки транспортних засобів України стандарту 1995 року. Приклади рядків, які підходять: «777-77XA», «12345XP», «229-06TC». Приклади рядків, які не підходять: «33-222XK», «321-04XX», «89012MI».
17.	Рядки, що містять коректні номери кредитних карток Visa, Mastercard або Простір. Приклади рядків, які підходять: «4012888888881881», «5105 1051 0510 5100», «9804 1286 0012 7543». Приклади рядків, які не підходять: «1234567812345678», «0000 0000 1111 2222», «401278684828183».
18.	Рядки, що містять час у форматі «ГГ:ХВ:СС». Приклади рядків, які підходять: 12:20:45, 23:00:54. Приклади рядків, які не підходять: 12:00, 14:11:67.
19.	Рядки, які містять коректні номери внутрішнього паспорта громадянина України, паспорта громадянина України для виїзду за кордон або коректний реєстраційний номер облікової картки платника податків. Приклади рядків, які підходять: «ММ450190», «FE126018», «000111222», «2005820864». Приклади рядків, які не підходять: «123456EE», «235768».
20.	Рядки, що містять коректні номери студентських квитків. Приклади рядків, які підходять: «ВК № 10330615», «ХА № 10450718», «ХА 10320416». Приклади рядків, які не підходять: «ХАI12345678», «1234567890».