

קלט/פלט וקבצים – File & Stream IO

העברת נתונים אל או מהתקן אחסון נחשבים פעולות קלט/פלט, לדוגמה: פעולות על קבצים הם פעולות קלט פלט. קריאה מקובץ היא פעולת קלט, וכתיבה לקובץ היא פעולת פלט.

התקן אחסון יכול להיחשב כל התקן בו יש נתונים או שניתן לשים בו נתונים, זה יכול להיות בתים בזיכרון של המחשב שלנו, זה יכול להיות דיסק מקומי שהוא התקן שנמצא בדרך כלל בתוך הקופסא של המחשב, זה יכול להיות התקן שמחובר למחשב מחוץ לקופסה כמו דיסק און קי, מצלמה שמחוברת בקאבל, מדפסת וכד' וזה יכול להיות דיסק בענן, או התקן שמחובר ע"י בלוטות ולא ע"י יציאת תקשורת פיזית (סריאלית או אחרת).

בדוטנט מרחב השמות System.IO מכילה מחלקות שיסייעו לנו לעבוד עם קבצים או מקורות אחרים שמזרימים מהם או אליהם נתונים. מרחב שמות זה מכיל מחלקות שמסייעות לקרא ולכתוב לקבצים, לכווץ קובץ ולחלצו,

קובץ הוא אוסף של בתים שנשמר בזיכרון קשיח (שלא נמחק עם כיבוי המחשב), יש לו שם, יש לו נתיב שבו הוא נמצא (באיזה דיסק, באיזה תיקיה ותת תיקיה). כשעובדים עם מחלקות המייצגות קבצים, ניגש אל הקבצים ע"י ציון המיקום שלהם ושם המלא. לעומת זאת, ישנם מחלקות המתייחסות לקבצים כמו אל מקורות מידע אחרים, ומתייחסים לקובץ כאוסף בתים, שניתן להזרים אליהם ומהם נתונים. אוסף המחלקות האלו משתמשים במילה Stream (זרימה).

Stream של נתונים, כאמור, יכול להיות מול אוסף בתים בזיכרון, אוסף של בתים בדיסק או מקורות מידע שמחוברים למחשב דרך רשת או בקאבל.

קבצים ותיקיות

המחלקה System.IO מכיל מחלקות המאפשרות לבצע אינטראקציה עם קבצים. ניתן למשל לקרא ולשנות באמצעותם את המאפיינים של הקבצי. לקבצים יש מאפיינים כמו: מחבר, סוג (סיומת) ועוד.

נתבונן בכמה מהמחלקות:

File - מחלקה המספקת אוסף מתודות סטטיות לעבודה עם קבצים (יצירה, העתקה, מחיקה, העברת מיקום, ופתיחת קבצים).

נשתמש בה כאשר יש לנו פעולות מול קבצים, ואין לנו עבודה חוזרת עם הקבצים איתם עובדים, כך שאין לנו צורך להחזיק מצביע לקובץ.

FileInfo - מחלקה המספקת מצביע לקבצים, יוצרים מופע של המחלקה ומצביעים לקובץ. מאפשרת פעולות דומות למחלקה File רק באמצעות מופע ומתודות ששייכים למופע, במקום המתודות הסטטיות.

גם במחלקה זו וגם במחלקה File ניתן להשתמש בכדי ליצור מופעים של FileStream מחלקה שמייצגת את העבודה השוטפת של קריאה וכתיבה מהקבצים.

Directory - מחלקה המאפשרת עבודה מול תיקיות באמצעות מתודות סטטיות. (יצירה, מחיקה, העתקה, העברה, ומעבר על הקבצים שהיא מכילה)

DirectoryInfo - מאפשרת עבודה עם התיקיות כנ"ל, באמצעות עבודה עם מופעים של המחלקה (ולא מתודות סטטיות). כך ניתן ליצור מצביעים לתיקיות שונות ולעבור עליהם במקביל עם עקרונות ה OOP.

Path - מחלקה המאפשרת לעבוד עם נתיבים של תיקיות ומיקומים של קבצים, ומסייעת לנו לעשות מניפולציות שונות על הנתיבים והשמות.

נ.ב. המחלקה נכתבה כך שתתמוך במערכות הפעלה שונות שאפשר להריץ עליהם את התוכנה שלנו. כשאנחנו כותבים קוד ב .NET מתקדם () ניתן להריצו עקרונית על לינוקס, ווינדויס ומערכות אחרות. צרת הגישה לקבצים ותיקיות שונה בכל מערכת הפעלה, Path יכול להוות פיתרון כשיש לנו אפליקציה שתרוץ על מערכות הפעלה שונות או כשלא ידוע מראש על איזה מערכת הפעלה התוכנה שלנו תרוץ. הנושא מחוץ ל Scope של הקורס, אבל נזכור שייתכן ונצטרך להשתמש גם במחלקות ייעודיות לקבצים ותיקיות כשמפתחים לפלטפורמות מסוימות.

נתיבים של קבצים

ב Windows שמות הדיסקים הן אות (בדרך כלל) ואחריהם נכתוב נקודותיים (:).

מפרידים בין שם הדיסק לתיקיה בין התיקיות עם קו נטוי (\).

מכיוון וקו נטוי מייצג תוי בקרה, המחרוזת עלולה להשתבש ונקבל שגיאה בזמן ריצה, לכן נשתמש בכפולות של קו נטוי או באופרטור @ לפני המחרוזת.

כדאי להקפיד כשמשתמשים בנתיבים ושמות לקבצים ותיקיות לשים את האופרטור @ לפני המחרוזת המייצגת את הנתיב.

"D:\NacmanFiles\MyFile.txt"

@ "D:\NacmanFiles\MyFile.txt"

@ "\127.0.0.1\NacmanFiles\MyFile.txt"

נתיבים יחסיים

קבצי טקסט

לעיתית קרובות נשתמש בקבצי טקסט לאחסון בהם מידע, הם נוחות וקלות לשימוש, וניתן לקרוא אותם גם עם עורכי טקסט למיניהם. זה מתאים כאשר המידע הוא קסטואלי, לא נשמר תמונה כקובץ טקסט, שכן במהות תמונה לא ניתן לבטא במילים. עם זאת, רשומה כמו "ספר" "סטודנט" "פציינט" שמכילים מידע על ספרים או על שירים, וכו' ניתן לתאר בצורה טקסטואלית. כלומר לשמור את הערכית של כל המאפיינים שלהם באופן טקסטואלי.

ברור שהמידע מאוחסן בצורה בינארית גם כשאנחנו מדברים על קבצי טקסט, אך כל עורך טקסט כמו גם התוכנה שלנו שקוראת את הקובצת, מאחסנת את הבתים בצורה שמייצגת טקסט. לדוגמה: A ערך ה ASCII (גם היוניקוד) שלו 65 בבינארית, זה נראה כך 01000001 (רק תוסיפו עוד אפסים משמאל). כשאנחנו כותבים לקובץ, מבחינתנו כתבנו A ובזיכרון סדר הביטים נראה כמו שהראינו. כשקוראים את הקובץ, עורך הטקסט וגם התוכנה שלנו, תקבל את זה כ A, כך הם מפרשים את סדר הביטים הנ"ל.

התוכנה שלנו, תיקח את הטקסט שהיא מקבלת מהקובץ ותמיר את השדות השונים לסוגים שבהם התוכנה צריכה אותם. בדרך כלל נשתמש בכלים שונים שיעזרו לנו בתהליך ההמרה ולא נמיר שדה שדה.

פורמטים נפוצים לשמירת נתונים (בקבצי טקסט)

כששומרים אוסף רשומות מסוג מסוים, נהוג לשמור את הקבצים בפורמטים מסוימים מוכרים מראש, אם כי כל אחד יכול להמציא מבנה ייחודי לקובץ שלו, בדרך כלל לא נעשה זאת.

Fixed Length Fields – בפורמט זה כל שורה מייצגת רשומה, נחליט מראש מה הגודל שיש להקצות לכל שדה, למשל, Id בגודל 4 תווים, שם פרטי 20 תווים, שם משפחה 20 תווים.

Comma Separated Value – CSV, בפורמט זה כל שורה מייצגת רשומה או שיש תו אחר שהוחלט שהוא מפריד בין הרשומות, כמו"כ יש תו שמחלטים שהוא מפריד בין הערכים השונים.

XML – פורמט טקסטואלי בה לכל שדה יש תיאור בתג לשדה ובתוכו הערך של השדה.

JSON – כמו ב XML יש תג שמייצג את השדה (שם השדה) ואחריו בה הערך. JSON חסכוני יותר מאשר XML.

קבצים בינאריים

קובץ בינארי נחשב בדרך כלל כמורכב מרצף של [בתים](#), כלומר [ספרות בינאריות](#) המאורגנות בקבוצות של שמונה. קובץ בינארי טיפוסי מכיל בתים המיועדים להתפרש שלא כתו טקסט פשוט.

קבצים בינאריים מסוימים מכילים "כותרת", קטעים של [metadata](#) המשמשים תוכנות מחשב לפירוש המידע שבקובץ. לדוגמה, קובץ [GIF](#) יכול להכיל מספר תמונות, והכותרת משמשת כדי לזהות ולהגדיר כל מקטע תמונה.

שמציגים קובץ בינארי בעזרת עורך טקסט רגיל, כל שמונה [סיביות](#) מפורשות כתו טקסט (במצב רגיל), והתוצאה תהיה טקסט חסר משמעות, בדרך כלל. כשפותחים קובץ בינארי על ידי תוכנה אחרת, ייתכנו תוצאות אחרות, למשל, תוכנה אחת תפרש את תוכן הקובץ כמספרים מ-0 עד 255 (טווח המספרים האפשרי בעזרת 8 סיביות) ותציג על המסך טור של מספרים לפי תוכן הקובץ. תוכנה אחרת תפרש את הקובץ כמכיל קודים לייצוג צבעים, ותציג על המסך תמונה המורכבת מצבעים לפי תוכן הקובץ.

עבודה עם FileStream

הרשאות גישה לקבצים

עבודה עם JSON

Readers and Writers

IDisposable & using Statement

תרגילי כיתה ובית

1. הדפס את שמות 10 התיקיות הראשונות שתמצא בדיסק המרכזי שלך (אם יש לך Windows זה כנראה C או D).
2. כתוב מתודה המקבלת שם תיקיה, המתודה תדפיס את שמות 3 הקבצים הכי גדולים שקיימים בה, ואת תאריך השינוי האחרון שלהם.
3. צור רשימת סטודנטים או ספרים, שמור בקובץ טקסט כ JSON את הרשימה.

4. שמור את הרשומות ב Fixed Length
5. שמור את הרשומות ב CSV (לסיום פתח את הקובץ עם תוכנת אקסל).
6. כתוב קוד שקורא את רשימת הסטודנטים (או הספרים) שיצרת בשאלה הקודמת.
7. איזה פורמט חסכוני יותר בזיכרון CSV או Fixed Length?
8. איזה חסרונות יש בפורמט Fixed Length?
9. מה היתרונות שיש בקובץ בינארי?
- 10.

קרדיטים: התכנים הכתובים בדרך כלל נאספים ממקורות שונים, חלקם במילים שחוברו לכבודכם, ובחלקם נבחר שלא להמציא את הגלגל מחדש ולהשתמש במלל שנלקח מהרשת. מאמר זה מתורגם מהאתר הרשמי של מייקרוסופט, וכן מכיל תכנים מויקיפדיה בעברית.