תוכן עניינים

[דוח שלב 0 1](#_Toc382231674)

[מבוא – מבני נתונים בסיסיים: 1](#_Toc382231675)

[המחלקה Disk 5](#_Toc382231676)

[המשתנים של המחלקה 5](#_Toc382231677)

[הפונקציות של המחלקה 5](#_Toc382231678)

[המחלקה Functions 6](#_Toc382231679)

[הפונקציות של המחלקה 6](#_Toc382231680)

[סכמתיות של מבנה התוכנית 6](#_Toc382231681)

# דוח שלב 0

בשלב זה של המיני- פרויקט, מגדירים ומיישמים מחלקה מתאימה לתיאור הדיסק.

## מבוא – מבני נתונים בסיסיים:

לשם הגדרת הדיסק, מוגדרים מספר מבנים המיצגים את מבנה האיברים בדיסק.

המבנהBaseSector

ראשית מוגדר מבנה האב **BaseSector** המייצגת את הסקטור אשר כל מבני הסקטורים השונים יורשים ממנה (מבנה הירושה הזה אינו נותן יתרון אמתי מבחינה תכנותית, אבל נכון יותר מבחינה רעיונית למבנה הדיסק, כי הרי אין הבדל פיזי בין הסקטור של ה DAT לסקטור של נתונים)

המבנה מכיל את השדות הבאים:

|  |  |  |  |
| --- | --- | --- | --- |
| שם השדה | טיפוס | משמעות | אורך השדה |
| sectorNr | Unsigned int | מספר הסקטור | 4 bytes |
| סה"כ גודל: | חישוב: 4=4 | | 4 bytes |

ישנם 5 מבנים שיורשים ממנו :

* **Sector** המיצג את מבנה הנתונים הבסיס הדפולטיבי של הדיסק,
* **VolumeHeader** המיצג את התווית של הדיסק,
* **RD\_Sector**  המיצג סקטור שהוא חלק מתקייה ראשית או משנית,
* **FileHeader** המיצג את הסקטור הראשון של קובץ (שייך לשלב 1),
* **Dat**המיצג את ה DAT של הדיסק.

המבנהSector

המבנה מייצג טיפוס נתונים פשוט בדיסק, ללא תכונות מיוחדות.

המבנה מכיל את השדות הבאים (מלבד sectorNr מהירושה):

|  |  |  |  |
| --- | --- | --- | --- |
| emptyArea | char[1020] | שמור לשימוש עתידי | 1020 bytes |
| סה"כ גודל: | חישוב: 1024=1020+4 | | 4 bytes102 |

המבנהVolumeHeader

מטרת המבנה לייצג את תווית הזיהוי של הדיסק

המבנה מכיל את השדות הבאים (מלבד sectorNr מהירושה):

|  |  |  |  |
| --- | --- | --- | --- |
| שם השדה | טיפוס | משמעות | אורך השדה |
| diskName | char[12] | שם זיהוי הדיסק | 12 bytes |
| diskOwner | char[12] | שם בעל הדיסק | 12 bytes |
| prodDate | char[10] | תאריך יצור הדיסק (ddmmyyyy) | 11 bytes |
| ClusQty | unsigned int | סה"כ יחידות הקצאה (clusters)  בדיסק | 4   bytes |
| dataClusQty | unsigned int | מספר יחידות הקצאה לנתונים בלבד. | 4   bytes |
| addrDAT | unsigned int | כתובת הסקטור שמכיל את ה-DAT | 4   bytes |
| addrRootDir | unsigned int | כתובת ה-cluster שמכיל את התיקייה הראשית (Root Directory) | 4   bytes |
| addrDATcpy | unsigned int | כתובת הסקטור שמכיל עותק שני של ה-DAT | 4   bytes |
| addrRootDirCpy | unsigned int | כתובת ה-cluster שמכיל עותק שני של התיקייה הראשית (Root Directory) | 4   bytes |
| addrDataStart | unsigned int | כתובת ה-cluster הראשון בדיסק המיועד לנתונים. | 4   bytes |
| formatDate | char[10] | תאריך פירמוט (ddmmyyyy) | 11 bytes |
| isFormated | Bool | האם כבר מפורמט? (כן / לא) | 1 byte |
| emptyArea | char[944] | שמור לשימוש עתידי | 945 bytes |
| סה"כ גודל: | חישוב: 1024=4+12+12+11+4\*7+11+1+945 | | 4 bytes102 |

המבנהDat

מטרת המבנה לייצג את הDAT של הדיסק.

המבנה מכיל את השדות הבאים (מלבד sectorNr מהירושה):

|  |  |  |  |
| --- | --- | --- | --- |
| שם שדה | טיפוס | משמעות | אורך השדה |
| DAT | DATtype | DAT | 200 bytes |
| emptyArea | char[820] | שמור לשימוש עתידי | 820 bytes |
| סה"כ גודל: | חישוב: 1024=4+200+820 | | 4 bytes102 |

המבנהDirEntry

מטרת המבנה לייצג איבר בדיסק; תיקיה או קובץ.

המבנה מכיל את השדות הבאים:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שם שדה | | טיפוס | משמעות | | אורך השדה | |
| Filename | | char[12] | שם הקובץ | | 12 bytes | |
| fileOwner | | char[12] | שם בעל הקובץ | | 12 bytes | |
| fileAddr | | unsigned int | כתובת הסקטור הראשון של הקובץ | | 4 bytes | |
| crDate | | char[11] | תאריך יצירת הקובץ | | 11 bytes | |
| fileSize | | unsigned int | גודל הקובץ, כמספר סקטורים | | 4 bytes | |
| eofRecNr | | unsigned int | מיקום "רשומת" ה-end-of-file (המספר הסידורי של מיקומה מהתחלת הקובץ) | | 4   bytes | |
| maxRecSize | | unsigned int | אורך רשומה מרבי | | 4 bytes | |
| actualRecSize | | unsigned int | אורך רשומה בפועל | | 4 bytes | |
| recFormat | | char[2] | סוג:  -          אם מדובר בקובץ נתונים, זה מסמל סוג רשומה כלומר, אורך קבוע או משתנה "F" : או  "V"  -          אם מדובר בתת-תיקיה, ערך השדה הזה יהיה האות "D". | | 2 bytes | |
| keyOffset | | unsigned int | Offset של התחלת המפתח בתוך הרשומה | | 4 bytes | |
| keySize | | unsigned int | אורך המפתח, כמספר בתים | | 4 bytes | |
| keyType | | char[2] | טיפוס נתונים של ערך המפתח:  "I" -  מספר שלם (int)  "F" – מספר ממשי  (float)  "D" – מספר ממשי כפול (double)  "C"  - מחרוזת תווים | | 2 byte | |
| entryStatus | | unsigned char | שדה זה מעיד על מצב הכניסה הספציפית בתיקייה. המצב יכול להיות אחד מתוך שלושה:  0        -  כניסה ריקה (empty): הכניסה עדיין לא הייתה בשימוש מאז שבוצע format  על הדיסק .  1        -  כניסה פעילה (active): הכניסה מייצגת קובץ קיים ופעיל.  2        -  כניסה לא פעילה (inactive): הכניסה מייצגת קובץ מחוק. | | 1 byte | |
| סה"כ גודל: | | חישוב: 72=12+12+4+11+4\*4+2+4\*2+2+1+ | | | Bytes72 | |

המבנים RD\_Sector ו RootDirectory

מטרת המבנה RootDirectory לייצג תיקיה.

המבנה מכיל מערך של שני מבנים מסוג RD\_Sector, כאשר RD\_Sector מייצג מערך של 14 איברים מסוג DirEntry כך ש RootDirectory מכיל סה"כ מערך בין 28 איברים המיצגים איברים בתקיה(קבצים או תיקיות משנה) בנוסף מועמס גם כ"כ אינדקס בכדי לרוץ באינטרציה על RootDirectory כך שיראה כמבנה יחיד.

המבנה RD\_Sector מכיל את השדות הבאים (מלבד sectorNr מהירושה):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שם שדה | | טיפוס | משמעות | | אורך השדה | |
| Dir | | DirEntry[14] | מערך איברים בדיסק | | 1008 bytes | |
| emptyArea | | char[12] | שמור לשימוש עתידי | | 12 bytes | |
| סה"כ גודל: | | חישוב: 1024=41008++12 | | | Bytes304 | |

המבנה RootDirectory מכיל את השדות הבאים:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| שם שדה | | טיפוס | משמעות | | אורך השדה | |
| sector1 | | RD\_Sector | מערך ראשון של איברים | | 1024 bytes | |
| sector2 | | RD\_Sector | מערך שני של איברים | | 1024 bytes | |
| סה"כ גודל: | | חישוב: 2048=1024\*2 | | | Bytes2048 | |

המבנהFileHeader

המבנה מיועד ליצג את התווית של קובץ כלשהו.

המבנה FileHeader מכיל את השדות הבאים (מלבד sectorNr מהירושה):

|  |  |  |  |
| --- | --- | --- | --- |
| שם שדה | טיפוס | משמעות | אורך השדה |
| fileDesc | dirEntry | העתק של כניסתו של הקובץ בתיקייה (file descriptor) | 72 bytes |
| FAT | DATtype | FAT | 200 bytes |
| emptyArea | char[744] | שמור לשימוש עתידי | 748 bytes |
| סה"כ גודל: | חישוב: 1024=4+72+200+748 | | Bytes1024 |

## המחלקה Disk

המחלקה מייצגת את המבנה הפיזי של הדיסק,

### המשתנים של המחלקה

המחלקה מכילה:

מערך מסוג Dat בשם dat המייצג את DAT .

איבר מסוג RootDirectory בשם rootdir המייצג את ספריית השורש של הדיסק

משתנה בוליני בשם mounted הנותן מידע האם הקובץ הותחל.

מצביע לקובץ הדיסק בשם. dskfl

משתנה מסוג unsigned int בשם currDiskSectorNr המייצג את המספר הסידורי של הסקטור בדיסק שכרגע בחוצץ של קובץ מסוים.

|  |  |  |  |
| --- | --- | --- | --- |
| שם שדה | טיפוס | משמעות | אורך השדה |
| vhd | VolumeHeader | תווית הדיסק | 1024 bytes |
| Dat | Dat | DAT הקובץ | 1024 bytes |
| rootdir | RootDirectory | תקיית השורש של הדיסק | 2048 bytes |
| mounted | bool | האם הדיסק מוצב או לא | 1 byte |
| dskfl | fstream\* | מצביע לסטרים לקובץ | 4 byte |
| currDiskSectorNr | unsigned int | המספר הסידורי של הסקטור בדיסק שכרגע בחוצץ של קובץ מסוים. | 4 bytes |
| סה"כ גודל: | חישוב: 2048=+2048+1+4+41024\*2 | | Bytes4105 |

## הפונקציות של המחלקה

המחלקה מכילה:

* בנאי ריק אשר רק מאתחל את איבר מסוג דיסק ללא שמירה פיזית וללא השמת ערכים בתווית.
* בנאי המקבל את שם הקובץ ושם ובעלים וכאשר נדרש מאתחל איבר מסוג דיסק שומרת את כל הערכים המתאימים בתווית, מאתחלת את הDAT והתיקיה הראשית. בנוסף היא שומרת את כל הדיסק בצורה פיזית על קובץ, שומרת את התווית של הדיסק התיקיה הראשית והDAT בתחילת הדיסק ועותק גיבוי בסופו ובנוסף היא מציבה סקטורים ריקים בכל הסקטורים. או רק מאתחלת את הדיסק הקיים.
* Createdisk – פונקציה המקבלת את שם הדיסק והבעלים, היוצרת את קובץ הדיסק ומציבה את שמו ושולחת לפונקצית המשך יצירת הדיסק, ולסיום סוגרת את הקובץץ
* Createdisk – פונקציה המקבלת את שם ההבעלים, ומסיימת לקבוע את הערכים של תווית הדיסק בנוסף מטפלת בקביעת הערכים של התיקיה הראשית ומאתחלת את הDAT עם הערכים הראשוניים הרלוונטים. לאחמ"כ הפונקציה שולחת לשמירה של הערכים.
* Savechanges – פונקציה המבצעת את השמירה בפועל של הדיסק, הפונקציה שומרת את התווית של הדיסק התיקיה הראשית והDAT בתחילת הדיסק ועותק גיבוי בסופו ובנוסף היא מציבה סקטורים ריקים בכל הסקטורים. או רק מאתחלת את הדיסק הקיים, כאשר היא מוודאת קודם כי יש צורך בשמירה על הדיסק והערכים לא שמורים כבר.
* Mountdisk – הפונקציה פותחת דיסק קיים ושומרת את התווית של הדיסק ואת הDAT באיברי המחלקה Disk הרלוונטים.
* Unmountdisk- הפונקציה סוגרת את קובץ הדימוי של הדיסק כאשר קודם לכן היא שומרת את הסקטורים שוב.
* Getdskfl- מחזירה מצביעה לקובץ.
* seekToSector- הפונקציה מקבלת int המיצג את הסקטור אותו יש צורת לכתוב/לקרוא ומעבירה את המצביע למקום זה.
* writeSector- מקבלת סקטור ומספר סקטור וכותבת את הסקטור ומיקום זה.
* writeSector – מקבלת סקטור ושומרת אותו למקום עליו מצביע המצביע.
* readSector – מקבלת סקטור ומספר סקטור וקוראת סקטור מהדיסק במקום המבוקש אל תוך הסקטור הניתן.
* readSector – מקבלת סקטור וקוראת סקטור מהמקום עליו עומד המצביע על תוך הסקטור שקיבלה.

# המחלקה Functions

המחלקה היא מחלקת עזר שמכילה את פונקציות עזר שאינם קשורים דווקא למחלקה מסוימת

## הפונקציות של המחלקה

* dateNow המקבלת מצביע למערך char ומציבה בו את התאריך הנוכחי.

# סכמתיות של מבנה התוכנית

