



תכנית הלימודים לעבודת גמר "מערכות טלפונים חכמים"
(מאחדת את החלופות בעבר – מערכות טלפונים חכמים תחת מערכת ההפעלה אנדרואיד (בסביבת
אקליפס בג'אווה), מערכות טלפונים חכמים תחת מערכת ההפעלה אנדרואיד בסביבת ויז'ואל סטודיו
בסי שארפ Xamarin (זמרון))

תכנית הלימודים להתמחות

"מערכות טלפונים חכמים"

מגמה 14.10

הנדסת תוכנה וסייבר

3	תקציר התוכנית.....
6	פרקי הלימוד בהתמחות.....
9	פרק 1 – סקירה כללית – אנדרואיד (Android).....
11	פרק 2 – תכנות מונחה עצמים.....
15	פרק 3 – אוספים.....
16	פרק 4 – חריגות.....
18	פרק 5 – מבוא ליישום/אפליקציה: פעילות/Activity.....
20	פרק 6 – ממשק משתמש/ UI-User Interface רכיבים ותבניות.....
23	פרק 7 – תכנות מונחה אירועים ושימוש במחלקה הפנימית.....
25	פרק 8 – מסרים Intents.....
28	פרק 9 – רכיבים מתקדמים: תפריט, תיבת דו שיח, הודעות, ה Status bar וטיפול באירועים.....
30	פרק 10 – משאבי האפליקציה Application Resources.....
32	פרק 11 – אחסון נתונים Data Storage.....
34	פרק 12 – תכנון וניתוח אפליקציה, שימוש ב UML, תיאור מסכים, תרשים זרימה.....
36	פרק 13 – ספקי תוכן Content Providers.....
38	פרק 14 – תהליכים ותהליכונים Threads & Processes.....
41	פרק 15 – שירות Service.....
43	פרק 16 – מקלט שידורים BroadCastReciever ו AppWidget.....
45	פרק 17 – טלפוניה.....
47	פרק 18 – רכיבים מתקדמים: מולטימדיה, מצלמה, חיישנים, GPS, מפות.....
49	פרק 19 – גרפיקה ואנימציה.....
52	פרק 20 – הנחייה והגשת עבודת הגמר.....

תקציר התוכנית

בחלופה זו התכנות יתבצע תחת מערכת ההפעלה אנדרואיד.

למעשה החשיבות היא בעיקר היכרות עם עקרונות העבודה תחת מערכת הפעלה זו.

ניתן לבצע את הפיתוח עצמו בצורות שונות, למשל בסביבת העבודה ויזואל סטודיו כשהתכנות נעשה ב-סי-שארפ (C#) או בסביבות עבודה אחרות כמו אקליפס (eclipse) או אנדרואיד סטודיו כשהתכנות נעשה בג'אווה. אולם כמו שקורה לעיתים בעולם מדעי המחשב ישנן אפשרויות נוספות ששכיחות משתנה במהלך הזמן.

למשל יתכן ותעדיפו להתשמש באקליפס ולהתקין תוסף ADT (Android Development Tools), או אולי תעדיפו את netbeans. ויתכן ותבחרו ב**קוטלין** (Kotlin) שעל פי מקורות שונים תהפוך לשפה "רשמית" של גוגל בתכנות לאנדרואיד.

(לגבי **קוטלין**, מוזמנים לעין ב <https://www.geektime.co.il/google-goes-kotlin-first>)

כאמור הפלטפורמה משנית כאן, למרות שחלק מהתכנים שיפורטו בהמשך יתמקדו בשתי הסביבות הראשונות שהוצגו שכן הן הנפוצות היום בקרב המורים. במספר מקומות בתוכנית הלימודים ירשם "בהתאם לסביבת העבודה העתידית (סי שארפ או ג'אווה)". חופש הבחירה הוא בידי המורה המלמד אבל בכפוף לדרישות תוכנית הלימודים.

בהתאם, לעיתים תמצאו בתוכנית הלימודים שמות על בסיס אחת השפות ללא שיכפול והצגת השם בשפה החליפית, אבל הכוונה ברורה (למשל עם נכתב GetData() לא תוצג המקבילה (getData()).

על פניו נראה שאנדרואיד נמצאת כאן כדי להישאר, אך כדאי לעקוב אחר השינויים בפופולריות מערכות ההפעלה לניידים לאורך 20 השנה האחרונות.

(סרטון המציג שינויים אלו נמצא ב <https://www.youtube.com/watch?v=MMyMB4zm9so>)

מטרתה העיקרית של החלופה תכנות תחת מערכת ההפעלה אנדרואיד היא לחשוף תחום זה בפני התלמידים מגיל צעיר, תוך שילוב וחשיפה להתקדמויות בעולם של תחום הטלפונים החכמים. הטלפון החכם היום נחשב לאחד מתחומי הפיתוח הגדולים בעולם, טלפון חכם נחשב למחשב לכל דבר, עם יכולות העולות על אלו של מחשבים ישנים ומאפיינים ייחודיים העולים על אלו של מחשבים עכשוויים כמו התוספות הקשורות לטכנולוגית התקשורת, חיישנים, מסך מגע, GSM, WIFI, GPS, BT ועוד.

מערכת ההפעלה אנדרואיד נבחרה להיות מערכת ההפעלה של מכשירים חכמים ע"י חברות הי-טק רבות, ומהן ענקיות הטכנולוגיה והתקשורת כמו Google, HTC, Intel, LG, Motorola, Toshiba, Sony Ericsson ו Vodafone, T-Mobile, Nvidia, Samsung OHA (Open Handset Alliance) תחת ברית OHA.

מטרות ביצועיות:

מטרת ההתמחות: להקנות לתלמידים כלים לפתח יישומים המשלבים ממשק משתמש גרפי על פי עקרונות הנדסת תוכנה, היישומים יפעלו על מכשירים חכמים וטאבלטים, תחת מערכת ההפעלה "אנדרואיד" של "גוגל".

- התלמיד יכיר ויסביר את העקרונות המתקדמים בפיתוח תוכנה בשפת תכנות מונחת עצמים.
- התלמיד יפתח תוכניות הכוללות תכנות מונחת אירועים.
- התלמיד יכיר, יסביר וידע להשתמש בתהליכים בתכנות מקבילי.
- התלמיד יסביר את מאפייני התוכנה שרצה על מכשירים חכמים.
- התלמיד יסביר ויפרש את הייחודיות של מערכת ההפעלה "אנדרואיד" של "גוגל" והמבנה שלה.
- התלמיד יסביר את עקרונות המכונה הווירטואלית ותפקידה וה- DAVLIK כדוגמה.
- התלמיד יכתוב ממשק משתמש הגרפי מתקדם בהקשר של ה"אנדרואיד"
- התלמיד יזהה רכיבים מתקדמים, רכיבי תקשורת וחיישנים במערכת כגון: מצלמה, מיקרופון, מקולרים, B, WiFi, GPS, Tooth, ועוד.
- התלמיד יכתוב אפליקציה למכשירים המופעלים ע"י מערכת ההפעלה "אנדרואיד" של "גוגל".

דרישות קדם לתוכנית לימודים זו:

1. היכרות טובה עם פרקים 1 – 6 בתוכנית הלימודים.
(מבוא, מושגי יסוד בתכנות, ביצוע מותנה, ביצוע חוזר, מבני נתונים סדרתיים, תכנות מונחה עצמים).
2. לימוד של יחידת הלימוד הבית ספרית (מס שאלון 899283).
במסגרת יחידת לימוד זו מומלץ לבחור את אחת מהחלופות הבאות:
 - מבוא לתכנות בסביבת אינטרנט.
 - תכנות מונחה אירועים (חלופה באישור מיוחד)
3. במקביל להוראת יחידה זו יש ללמוד את חלקה השני של תוכנית הלימודים במדעי המחשב – מבני נתונים (מחליף את עיצוב תוכנה).
במסגרת פרק הבחירה בתוכנית הלימודים רצוי לבחור בחלופה "תכנות מונחה עצמים".

המלצה לתהליך ההוראה

מוצע ללמד בגישה האינדוקטיבית – הסקה המפרט אל הכלל. דהיינו שימוש בדוגמה, שינוי בקוד הדוגמה, הצגת התוצר והבנת הקשר הסיבתי.

חלק נכבד מהתרגול יעשה כשיעורי בית ובמעבדה. ההנחה היא שסביבות הפיתוח המתאימות מותקנות במחשבי התלמידים ומאפשרת להם עבודה עצמאית תוך שימוש ברשת האינטרנט (בין אם חיפוש חופשי, שימוש במדריכים או אתרים המאפשרים הרצת קוד ברשת).

יש לנסות לאתגר את התלמידים ולדרוש מהם לבצע בבית משימות אשר לא ניתנה להם דוגמה בכיתה אבל הנושא בהם קרוב תוכנית למה שנדון בכיתה. שיטה זו תקנה לתלמידים הרגלים של שימוש באינטרנט בחיפוש אחר מידע, דוגמאות ואף מחלקות. הרגלים אלו יעזרו להם בפיתוח העתידי.

רצוי לדרוש הגשת מטלות ביניים אותן ניתן לבדוק באופן מדגמי. מטלות הביניים יספקו אפשרות מעקב אחר דרכי התנהלות התלמיד ויכולות הלמידה שלו ויעזרו בהמשך בניהול הפרויקט.

המלצה להנחיית הפרויקט

רצוי שבבואו ליצור את הפרויקט הסופי יהיו לתלמיד תוצרי ביניים שעליהם כבר עבד בעבר והוא יכול לשחזר את מימושם במהירות. לכן מומלץ לדרוש מהתלמיד לשמור העתקים של עבודותיו ברשת ב GIT או DRIVE כך שיהיו נגישים בכל זמן גם למורה וגם לתלמיד.

רצוי לתכנן את הפרויקט על בסיס לוח זמנים מוגדר, בצורה מודולרית וספירלית כך שבכל זמן תהיה אפשרות להציג התקדמות בתת נושא שעובדים עליו ובהמשך לשלבו בתוצר הקיים. כך שהתלמיד צריך להיות מוכן להציג תוצר "סופי לעת עתה" בכל זמן.

לרוב, התלמידים הניגשים לפרויקט זה מכוונים לפרויקט בהיקף של 5 יח"ל. ההגדרה אם הפרויקט יהיה בהיקף של 3 או 5 יחידות לימוד תהיה בהתאם לתכנים שיכללו בפרויקט פירוט התכנים בנספח ההערכה.

חלוקת השעות הרצויה

תוכנית לימודים זו מתוכננת ל 15 ש"ש (שעות שבועיות) ומתוכננת להיפרש על פני שנתיים. מטבעה היא משלבת שעות התנסות ושעות עיוניות. בדרך כלל היחס בין העבודה המעשית לעיונית הוא של 1 ל 2. כלומר 33% עיוני ו 67% מעשי.

חלוקה אפשרית היא של 180 שעות הוראה בכיתה יא - 6 ש"ש (עיוני + מעשי) ו 270 שעות הוראה בכיתה יב - 9 ש"ש (עיוני + מעשי) אולם בתי ספר יכולים לבחור בחלוקה אחרת בהתאם למבנה השנה האקדמית כפי שהיא נלמדת בהם. בכל מקרה סכום השעות צריך להיות בהתאם לסך כל השעות המתוכננות.

יש לבצע הערכה לפרויקט במהלך הלמידה ולא רק בהתאם לתוצר הסופי. זו תהיה הערכה מעצבת ותאפשר לתלמיד להתאים את אופי, מורכבות והיקף הפרויקט ליכולותיו וללוח הזמנים המתאים לו.

רצוי לאפשר לתלמידים לאתגר את עצמם. בהתאם לבחירת המורה לאפשר להם ללמוד תכנים נוספים מעבר לתכני החובה.

פרקי הלימוד בהתמחות

תכנים אלו מאורגנים על בסיס סדר הלמידה כפי שהיה נהוג בתכניות תכנון ותכנות טלפונים חכמים בעבר. לשיקול המורים ניתן לשנות את סדר ההוראה בהתאם להתקדמות הכיתה בתכני מקצוע מדעי המחשב הנלמד במקביל ו/או שיקולים רלוונטיים נוספים. למרות שהתוכנית תוגדר על בסיס טכנולוגיות מסוימות כפי שתואר בהקדמה, ניתן להפעילה בכל טכנולוגיה הפועלת לפי אותם עקרונות מנחים!!

פרק מס'	נושא לימוד	פירוט התכנים	שעות עיוני	שעות מעשי
1	סקירה: מה זה ה"אנדרואיד"? Android? מה סביבת העבודה.	סקירה כללית היסטוריה, ארכיטקטורה וכו', היכרות עם מושגים	1	2
2	תכנות מונחה עצמים	הורשה, פולימורפיזם, מחלקות, עצמים, ממשקים.	12	24
3	אוספים	אוספים גנריים באנדרואיד והתאמה לשפת המימוש	2	4
4	חריגות	טיפול בחריגות	1	2

6	3	פעילות / Activity אבני יסוד האפליקציה, מבנה הפרויקט, קובץ ההגדרות והמשאבים מעגל חיי הפעילות.	מבוא ליישום/אפליקציה:	5
20	10	רכיבים ותבניות XML מקוד (סי- שארפ או ג'אווה) בהתאם לבחירה, ורכיבי ממשק. סידור הממשק ע"י תבניות. שמירת מצב הפעילות בשינוי תצורה.	ממשק משתמש UI-User :Interface	6
6	3	טיפול באירוע לחיצה בשיטת XML בניית מאזינים: פעילות מממשק, מחלקה פנימית ויצירת מאזין בשורה.	טיפול אירועים, שימוש במחלקה פנימית וב C# גם sdelegate	7
8	4	משמעות המסרים וסוגיהם מסנני מסרים פעילות למטרת תוצאה	מסרים Intents	8
10	5	תפריטים, תיבות דו שיח, הודעות	תפריט, תיבת דו-שיח, Status bar הודעות וטיפול באירועים	9
6	3	מבנה תיקיית המשאבים וסוגיהם יצירת וארגון סוגים שונים מהמשאבים	משאבי האפליקציה Application Resources	10
24	12	אחסון העדפות משותפות Shared Preferences אחסון פנימי, אחסון חיצוני SQLite Database	אחסון נתונים Data Storage	11
8	4	ניתוח בשיטת UML, תיאור מסכים, תרשים זרימה, שרטוט ישויות .	תכנון וניתוח אפליקציות,	12
120	60		סך הכול שעות לימוד לכיתה יא	
16	8	משמעות ספקי התוכן וה URI. שימוש בספקי תוכן, שאלות	ספקי תוכן Content Providers	13
30	15	התכנות המקבילי Thread תהליכים Processes AsyncTask או Handler	תהליכים וההליכונים (Threads & Processes)	14
16	8	סוגי השירות, ומעגל החיים . בניית שירות כהרחבה של IntentService Service	שירות Service	15

		תזמון שירות ושירות מחזורי שירות חזית		
14	7	המחלקה BroadcastReceiver, מקלט, רישום, שליחה וקליטת מסר. שירותי מערכת, הרשאות ואבטחה. App Widgets	מקלט שידורים BroadCast App Widgets ו Reciever	16
16	8	TelephonyManager ו PhoneStateListener זיהוי שיחה יוצאת. קבלת ושליחת SMS	טלפוניה	17
24	12	סוגי החיישנים. אופן השימוש בחיישנים ע"י מנהל החיישנים וההאזנה להם. מפות, מדיה.	רכיבים מתקדמים: מולטימדיה, מצלמה, חיישנים, GPS, מפות.	18
16	8	ציור: XML, View, SurfaceView אנימציה: AnimationDrawable, SurfaceView, SurfaceHolder	גרפיקה ואנימציה.	19
48	24	ההבדל בין הרמות, מחקר וניתוח, קביעת לוח זמנים, ביצוע הפרויקט, תיק פרויקט	הנחיה והגשת עבודת גמר	20
180	90		סך הכול שעות לימוד יב	

פרק 1 – סקירה כללית – אנדרואיד (Android)

הקדמה

פרק זה מטרתו להציג לתלמידים את הרקע להתמחות טלפונים חכמים. מעבר למשמעות המילה וההיסטוריה של מערכת ההפעלה חשוב להציג את החלק שהיא תופסת בעולם פיתוח האפליקציות לטלפונים חכמים ואת החשיבות של היכרות עם הכלים לפיתוח תוכניות תחת מערכת הפעלה זו.

מטרות כלליות:

היכרות עם מערכת הפעלה אנדרואיד.
היכרות עם סביבת העבודה שנבחרה למימוש הפרויקט.

מטרות ביצועיות

1. התלמיד ידע אינפורמציה כללית על מערכת האנדרואיד.
2. התלמיד יכיר היכרות כללית של שוק האפליקציות של מערכת האנדרואיד (Android Market).
3. התלמיד ידע את מרכיבי מערכת האנדרואיד.
4. התלמיד יכיר ויתנסה בהתקנת SDK Manager.
5. התלמיד יכיר ויתנסה ביצירת סימולטור אנדרואיד ע"י AVD Manager.
6. התלמיד יתנסה במערכת האנדרואיד ע"י ה AVD.
7. התלמיד יכיר את מבנה ארכיטקטורת אנדרואיד.
8. היישום באנדרואיד: קובץ ה APK.
9. התלמיד יכיר גרסאות שונות וההבדלים המהותיים ביניהן.
10. סביבות קיימות – למשל זמרין (Xamarin) או אקליפס (Eclipse) ואחרות.
11. היכרות, התקנה ושימוש ב Android SDK.
12. דגשים נוספים בהתאם לסביבת העבודה העתידית (סי שארפ או ג'אווה).

מושגים והכוונה

1. ארכיטקטורת אנדרואיד.
2. SDK
3. APK
4. VM ו VMIDa.vik
5. Libraries
6. Framework

7. Applications.
8. Linux Kernel.
9. Runtime-Core Library.
10. מושגים האופייניים לסביבת העבודה העתידית כמו onomM...

דרכי הוראה:

בהתאם למטרת הפרק ניתן להעמיק היכרות על ידי מצגות והפנייה לסרטונים. העובדה שהמערכת מתפתחת כבר מעל לעשור וריבוי המושגים החדשים, מאפשרים גיוון בהוראה כמו למשל לחלק מושגים לתלמידים שילמדו עליהם בעצמם ויציגו לכיתה תכנים מעניינים מעבר למוצג בכיתה. (כמו מקור השם לינוקס, מהי תוכנה חופשית, מה הקשר ליוניקס, מה מקור השם vikiDa ועוד) או לפתוח קובץ APK בתוכנת ZIP, וכו'.

דרכי הערכה אפשריות לפרק זה:

1. כתיבת בחינת ידע כללי.
2. מטלות אישיות – הרחבה על מושג

חלוקת שעות:

נושא	עיוני	התנסות
סקירה כללית היסטוריה, ארכיטקטורה וכו', היכרות עם מושגים והיכרות והתנסות ב AVD Manager ו SDK Manager	1	2

פרק 2 - תכנות מונחה עצמים

הקדמה

פרק זה מתבסס על ההנחה שהתלמידים למדו תכנות מונחה עצמים ובמקביל לומדים זאת במסגרת פרק "מבני נתונים" שבתוכנית הלימודים. למרות זאת יש להציג את התכנות בגישת OOP (Object Oriented Programming) מבלי תלות בהוראה המקבילה. רצוי לתאם או לפחות להכיר את התכנים כפי שנלמדו בתוכנית הלימודים של המקצוע מדעי המחשב.

מטרות כלליות

הכרות עם מונחי יסוד בתכנות מונחה עצמים בהתאם למימוש העתידי בחלופה. הורשה, פולימורפיזם, ממשקים, מחלקה מופשטת, המרה כלפי מעלה וכלפי מטה, דריסת פעולות, זימון פולימורפי.

מטרות ביצועיות

1. התלמיד יסביר את המושגים מחלקה, עצם/מופע, קשרים בין עצמים ותקשורת בין עצמים.
2. התלמיד יזהה ויהבדיל בין ירושה להרכבה בין מחלקות.
3. התלמיד יבצע ירושה ממחלקה קיימת.
4. התלמיד יבצע ויסביר ביצוע ירושה.
5. התלמיד יסביר את מהות ותפקיד הממשקים (לדוגמה – מאזינים).
6. התלמיד יבצע יצירת ממשקים ושימוש בהם.
7. התלמיד יסביר את משמעות המחלקה המופשטת והשיטות המופשטות.
8. התלמיד יתאר ע"י תרשים UML את היחסים האפשריים בין המחלקות ובין המחלקות והממשקים.
9. התלמיד יתנסה בשימוש בפולימורפיזם וזימון פולימורפי: בשימוש בירושה, או במימוש ממשק.

מושגים והכוונה:

הורשה:

1. הורשה כמייצגת חלוקה היררכית טבעית-אנושית: תהליך של מיון וסיווג היררכי. ייצוג היחס "סוג של" (is a).
2. הורשה בסיסית שארץ ובג'אווה (בהתאם לחלופה)
 - i. **סי שארץ** (single inheritance) – התוספת נקודתיים בכותרת המחלקה, מחלקת-על ותת-מחלקה, הרשאת הגישה מוגן – protected.
 - א. הורשה – מה עובר? שיטות בונות, העברת פרמטרים למחלקת אב שיטה רגילה ולגבי קונסטרוקטור.
 - א. דריסה (הגדרה מחדש) של שיטות – virtual overriding
 - ii. **ג'אווה** (single inheritance) – התוספת extends בכותרת המחלקה, מחלקת-על (super class) ותת-מחלקה (sub class), הרשאת הגישה מוגן – protected.
 - א. הורשה – מה עובר? שיטות בונות, השימוש ב-super לגבי שיטה רגילה ולגבי קונסטרוקטור.
 - א. דריסה (הגדרה מחדש) של שיטות – overriding.
3. הורשה מהמחלקה הראשונה – Object.
4. הורשה כתומכת ברעיון הסתרת מידע, שימוש חוזר בקוד.
5. פולימורפיזם – רב צורתיות: היתרון, העוצמה של הרעיון והשימוש בו.
6. זימון פולימורפי של שיטות, המרה למעלה (up casting) והמרה למטה (down casting).
7. האופרטור instanceof, (ג'אווה) תוך הדגשת צמצום השימוש בו לטובת רעיון הפולימורפיזם.
8. האופרטור is, (סי-שארץ) תוך הדגשת צמצום השימוש בו לטובת רעיון הפולימורפיזם.
9. מחלקות עוטפות – הדרך להפוך טיפוסים בסיסיים לאובייקטים.

ממשק interface:

10. ממשק כמגדיר התנהגות ומייצג את היחס "מתפקד כ-" (לעומת "סוג של" בהורשה).
11. ממשק אינו מחלקה: הוא מגדיר טיפוס אך אינו משמש כתבנית ליצירת עצמים.
12. הממשק כמגדיר חוזה. כל מחלקה המעוניינת לממש את הממשק – חייבת לממש את כל המפורט בחוזה. חובות וזכויות המחלקות הממשות.
13. הממשק בסי-שארץ: תוספת הנקודתיים לכותרת מחלקה המממשת את הממשק.
14. הממשק בג'אווה: תוספת ה-implements לכותרת מחלקה המממשת את הממשק.
15. הורשה בין ממשקים. מימוש מרובה של ממשקים.
16. ממשקים בשירות הפולימורפיזם.

מחלקות מופשטות:

17. הגדרת הצורך בהגדרת מחלקות מופשטות: המחלקה המופשטת כמגדירה רעיון שאינו ניתן למימוש וליצירת עצמים בשלב נתון.
18. המחלקה המופשטת: כותרת המחלקה. מימושים חלקיים. הגדרת שיטות מופשטות. שיטות בנות.
19. המחלקה המופשטת כמגדירה חוזה. כל מחלקה המעוניינת לממש את המחלקה – חייבת לממש את כל המפורט בחוזה. חובות וזכויות המחלקות המממשות. הגדרה מחדש במחלקות הירושות.
20. מחלקות מופשטות בשירות הפולימורפיזם.
21. תרשים UML לתיאור מחלקות, מחלקות מופשטות וממשקים והקשרים האפשריים ביניהם.

דרכי הוראה:

יש לבדוק את מידת המיומנות של התלמידים בתכנות מונחה עצמים. בהוראת הורשה מומלץ להתחיל ע"י דוגמאות קרובות לתפיסת התלמידים כמו היחס בין בעלי חיים, בעלי חוליות, חסרי חוליות, יונקים, עופות, דגים וכו'. מומלץ להשתמש במצגות היחידה החמישית, תכנות מונחה עצמים, של האוניברסיטה העברית.

יש לתרגל עם התלמידים זיהוי היחס בין מחלקות ע"י שימוש "סוג של" ו "מורכב מ", ולתאר את היחס ב UML.

יש לתרגל עם התלמידים את התחביר לכתיבת ההורשה לפחות פעם אחת, ולתת להם תרשים UML המכיל יחסי הורשה והרכבה ולבקש מהם לממש אותו במחברת ובמחשב. אחרי המימוש במחשב לתת לתלמידים ליצור מחלקה עם פעולה ראשית לביצוע בדיקה למה שנכתב, ולהתנסות מעשית בהמרה כלפי מעלה (פולימורפיזם) והמרה כלפי מטה ומשמעותה.

לחזור על הנושאים תוך שימוש שמחלקה מופשטת כגון המחלקה צורה והפעולה המופשטת חשב שטח. לבקש מהתלמיד ליצור מחלקות למלבן, ריבוע ומעגל כהרחבה של המחלקה צורה ולבצע דריסה לפעולת חישוב השטח.

בשלב זה אפשר להתנסות באופן מעשי ומוחשי את הזימון הפולימורפי על מערך של צורות. לגבי ממשקים – אפשר להגדיר ממשק בשם AreaAble שמכיל פעולת ממשק לחישוב שטח לבקש מהתלמיד ליצור מחלקות למלבן, ריבוע ומעגל שמממשים ממשק זה ולבצע דריסה לפעולת חישוב השטח.

בשלב זה אפשר להתנסות באופן מעשי ומוחשי את הזימון הפולימורפי על מערך מטיפוס AreaAble.

דרכי הערכה אפשריות לפרק זה:

1. מבחן או תרגיל מעבדה בו יתבקש התלמיד לממש על המחשב תרשים UML וממשק מחלקות נתון, ומחלקת בדיקה תוך רישום דרישות ברורות. למעבדה יצורפו שאלות הבנה אותן ניתן יהיה לבדוק על ידי מימוש במחשב.

חלוקת שעות:

נושא	עיוני	התנסות
חלק מהתכנים ילמדו במקצוע המוביל, חשוב לבדוק מהם כדי למנוע חזרה ולאפשר הרחבה העמקה בהתאם לנדרש בתכנים הבאים: הורשה, פולימורפיזם, מחלקות, עצמים, ממשקים	12	24

פרק 3 - אוספים

הקדמה

אנדרואיד תומכת במגוון אוספים גנריים, שכולל מחלקות, מחלקות מופשטות וממשקים למימוש אוספים. פרק זה נועד להכיר אוספים אלו ולהבין את אופן המימוש שלהם לפי הצורך.

מטרות כלליות

התלמיד יכיר את מבני הנתונים והאוספים המובנים במערכת האנדרואיד וההבדלים ביניהם.

מטרות ביצועיות:

1. התלמיד יסביר את התפקידים של האוספים וההבדלים ביניהם.
2. התלמיד ישתמש באוספים הגנריים.
3. התלמיד יכתוב קוד שיבצע: הוספה, חיפוש, עדכון ומחיקה מכל סוג של האוספים.

מושגים והכוונה

יש לבחור את האוספים המתאימים לשפת המימוש (סי-שארפ או ג'אווה)

1. $\text{List} < E$ - אוסף שממש את List ומתנהג כמערך. מבנה דומה $\text{Vector} < E$.
2. $\text{Dictionary} < K, V$ - מחלקה המממשת את הממשק מפה, המחזיקה את הנתונים בשיטת מפתח/ערך.
3. $\text{ArrayList} < E$ אוסף שממש את List ומתנהג כמערך. מבנה דומה $\text{Vector} < E$. כמו ArrayList שמוגן גישה ע"י synchronized
4. $\text{HashMap} < K, V$ - מחלקה המממשת את הממשק מפה, שמחזיקה את הנתונים בשיטת מפתח/ערך.

דרכי הוראה:

תרגול על ידי בנייה של מחלקה שמנהלת נתונים, למשל: מחלקה המנהלת ספר טלפונים. יש ליישם את הדוגמה בעזרת שילוב האוספם המתאימים. ולעמוד על יתרונות וחסרונות כל יישום.

דרכי הערכה אפשריות לפרק זה:

מטלת ביצוע, לניהול נתונים לפי בחירת התלמיד כגון: ניהול עבודות בית, ציוני תלמיד (תעודה), נוכחות תלמידים, ציוני תלמידים (כיתה) ועוד.

חלוקת שעות:

נושא	עיוני	התנסות
אוספים גנריים באנדרואיד והתאמה לשפת המימוש	2	4

פרק 4 - חריגות

הקדמה

חריגות הן חלק בלתי נפרד מעולם התכנה, בשלב זה התלמיד כבר מודע לחלוקה הבסיסית לשלושה סוגי שגיאות – קומפילציה, לוגיות, וזמן ריצה. הדגש בפרק זה הוא מניעת שגיאות שגורמות לקריסת היישום. ומתן דגשים לסוגי החריגות השונות והטיפול בהן.

מטרות כלליות

הבנה שבתכנון ובניית תוכנה יש להתייחס גם לתקלות ולא רק לזרימה תקינה של התוכנית היא חשובה על מנת לבנות תוכניות אמינות והיכרות שגיאות זמן ריצה, וטיפול בחריגות

מטרות ביצועיות:

1. התלמיד יסביר את קיום ומשמעות החריגה.
2. התלמיד יבצע טיפול בחריגה ע"י תפיסתה באמצעות הבלוקים של try, catch, ו finally וידע להסביר את משמעות מילים שמורות אלו.
3. התלמיד יטפל בחריגה ע"י זריקה של חריגה באמצעות throw.
4. התלמיד יסביר במילים שלו, מתי כדאי או הכרחי לטפל בחריגה ע"י try, catch, finally או זריקתה ע"י throw.
5. התלמיד יזהה את הפעולות הקיימות ב API שזורקות חריגה.

מושגים והכוונה:

1. חריגות Exception מה היא חריגה: שגיאה מול חריגה.
2. חריגה היא עצם, המחלקה Exception.
3. טיפול בחריגות בסביבת העבודה:
 - i. throw.
 - ii. try.
 - iii. catch.
 - iv. finally.
4. חריגות מוגדרות מראש, כל החריגות הן סוג של Exception.

דרכי הוראה:

רצוי לבצע חריגות מכוונות כמו: גלישה מתחום מערך, חלוקה באפס, גישה להפניה שהינה null.

להתנסות ולזהות איך המערכת והיישום מגיבים ללא טיפול בחריגות ואחר כך להוסיף טיפול ע"י try, catch, finally ולעמוד על ההבדלים והתוצאות. לבצע טיפול באחת החריגות ע"י זריקה של לפחות 2 רמות עד להגעה למחלקה שיכולה להשתמש בממשק משתמש גרפי כדי להודיע על החריגה ואופן הטיפול בה.

דרכי הערכה אפשריות לפרק זה:

1. בחינה עיונית – ניתוח קוד שיוצר שגיאה לכידת השגיאה ותיאור התוצאה.
2. בחינה מעשית – כתיבת קוד עם סיכוי של חריגות מסוגים שונים, ושימוש בפעולות שזרקות חריגות, התלמיד יכתוב ויטפל בחריגות האפשריות ובחריגות שנזרקות ע"י הפעולות הנתונות.

חלוקת שעות:

התנסות	עיוני	נושא
2	1	טיפול בחריגות

פרק 5 – מבוא ליישום/אפליקציה: פעילות/Activity

הקדמה

יישום מובייל, או יישומון או אפליקציה הוא מונח שכיום מוכר לכולם. האפליקציות באנדרואיד יכולות להיכתב בשפת **סי-שארפ** או ב**ג'אווה** (וגם בשפות אחרות שאינן מפורטות בתוכנית לימודים זו) ועוברות הידור ע"י Android SDK. והתוצר מכיל את כל הקוד המקומפל וקבצי הנתונים והמשאבים בחבילה אחת בקובץ דחוס עם הסיומת apk. שהינו קיצור ל. Android package הפעילות או ה Activity מהרכיבים החשובים של היישום. הפעילות מייצגת מסך ממשק משתמש גרפי יחיד מהאפליקציה. אפליקציה יכולה להכיל מספר פעילויות. לכתובת פעילות צריך לבצע הרחבה (ירושה) למחלקה Activity

מטרות כלליות

הבנה של מהות היישום, היכרות עם המבנה, אבני היסוד, המשאבים וקובץ המניפסט AndroidManifest.xml. מעגל חיי הפעילות, המתודות המופעלות במעגל חיי הפעילות.

מטרות ביצועיות:

1. התלמיד יסביר באופן כללי את אבני היסוד של היישום, קיום קובץ הגדרות חיוני ומשאבי היישום.
2. התלמיד יצור פרויקט אנדרואיד בסביבת הפיתוח שנבחרה: eclipse, visual studio, או אחרת
3. התלמיד יבחין ויזהה את מבנה היישום בסביבת הפיתוח שנבחרה.
4. התלמיד יזהה את תפקיד הפעילות והבנת מעגל חייה כולל המעבר ממצב למצב, ותפקיד המתודות המופעלות בכל מעבר.
5. התלמיד יתנסה בזיהוי הפעולות המופעלות והמעבר ממצב למצב.

מושגים והכוונה:

1. אבני יסוד האפליקציה:

Service, Activity, Broadcast receiver, Content provider, Intent. הרכיבים המרכזיים שדרכן כותבים את האפליקציה. נחשבים לנושאים המרכזיים ויוסברו בהרחבה בהמשך. האפליקציה שחיים בה רכיבים אלו נקראת הקשר היישום (Application Context), כך מאפשרת שיתוף נתונים ביניהם.

2. הגדרות האפליקציה:

מבוא למבנה קובץ המניפסט AndroidManifest.xml. המכיל הגדרות עבור הפעלת האפליקציה המציניים את היכולות, דרישות והרשאות.

3. משאבי האפליקציה:

משאבי האפליקציה. Resources, הם צרכים חיצוניים שנצטרך להם ביישום שלנו, כגון תמונות, מחרוזות, צבעים, סגנונות ועוד. כל אלה אמורים להימצא בתיקה המתאימה תחת תיקיית המשאבים בפרויקט/res/".

4. מעגל חיי הפעילות:

לכל פעילות מעגל חיים, ומספר מצבים שהיא עוברת דרכם במהלך חייה, מרגע היצירה עד הסיום. מתחילה ממצב האתחול Starting ומסתיימת במצב מושמד Destroyed, וביניהם יכולה להיות במצב ריצה Running (נקרא גם Resumed), מושהה Paused או מופסק Stopped.

5. מתודות/פעולות מעגל חיי הפעילות:

(onCreate(), onStart(), onResume(), onPause(), onStop(), onDestroy())

מתודות שתפקידן להעביר את הפעילות ממצב למצב, הפעלת מתודות אלו במהלך ריצת היישום מתבצע כתוצאה מצורכי המערכת ומהאינטראקציה עם המשתמש. כדי לממשן צריך לדרוס אותן בהרחבת המחלקה Activity.

דרכי הוראה:

לדון באבני היסוד ותפקידם תוך מתן דוגמאות. להנחות את התלמידים ביצירת פרויקט ולעקוב אחרי מבנה הפרויקט בסביבת העבודה שנבחרה. יועבר לתלמידים תרשים מעגל החיים של הפעילות שכולל את המצבים ושמות המתודות המופעלות בכל מעבר ממצב למצב. חשוב שתתבצע דוגמה מעשית תוך שימוש בפלט של ה Logcat לזיהוי זימון פעולה מסוימת במעגל חיי הפעילות.

דרכי הערכה אפשריות לפרק זה:

1. בחינה מעשית של יכולת התלמיד ליצור פרויקט ולזהות את מרכיביו.
2. בחינה בכתב על מעגל החיים של הפעילות והיתכנות כל מצב.
3. בחינה במעבדה בה מתבקש התלמיד לבצע דרישות הדפסה מסוימת במעבר ממצב למצב ולהבחין שהינו יכול לזהות מתי צריכה לפעול כל פעולה.

חלוקת שעות:

נושא	עיוני	התנסות
אבני יסוד האפליקציה, קובץ ההגדרות	1	2
מבנה הפרויקט, קובץ ההגדרות והמשאבים	1	1
מעגל חיי הפעילות.	1	3

פרק 6 – ממשק משתמש / UI-User Interface רכיבים ותבניות

הקדמה

ממשק משתמש גרפי באנדרואיד מורכב מרכיבים גרפיים שנקראים View ו- ViewGroup. כל הרכיבים הגרפיים הם תתי מחלקה של View שנקראים גם וידג'טים (widgets). לשימוש ברכיבי ה UI באנדרואיד ניתן בשתי אפשרויות. הראשונה "תכנותית" (ע"י קוד סי-שארפ או ג'אווה) והשנייה הצהרתית ע"י קובץ XML.

מטרות כלליות

שימוש ברכיבי UI, שמירת מצב הפעילות והטיפול בשינוי תצורה. תבניות UI ב XML לסידור רכיבים, קישור עם הפעילות, רכיבי UI שימושיים והפלטה של סביבת העבודה (כמו visual studio או eclipse).

מטרות ביצועיות:

1. התלמיד יבנה ממשק משתמש גרפי ע"י תבנית XML.
2. התלמיד יבצע שימוש בתבנית ה UI מקוד (סי שארפ או ג'אווה) כמו הצגת התבנית, קבלת הפניות על רכיבי הממשק, ועוד.
3. התלמיד יפעיל חלק מהרכיבים הגרפיים ל UI.
4. התלמיד יאתר רכיבים שלא למד וישתמש בהם ע"י ממשק מתאים.
5. התלמיד ישתמש ברכיבי ממשק משתמש גרפי פשוטים שיוצר אותם ע"י קוד (סי שארפ או ג'אווה).
6. התלמיד יבצע שיטות לשמירת מצב (נתוני) הפעילות אחרי ביצוע שינוי על הצגה כגון סיבוב המכשיר, הצגת והעלמת מקלדת וכו'.

מושגים והכוונה:

1. ממשק משתמש (UI) ע"י קובץ תבניות XML ה- XML Layouts.
2. שימוש במשאב תבנית XML מקוד (סי שארפ או ג'אווה).
3. מאפיינים ופרמטרים בקובץ תבניות XML.
4. Form Widgets: ובה רכיבים של הצגת טקסט, מקשים לסוגיהם, ציר התקדמות, סרגל דירוג וכו'.
5. שדה טקסט: קיימים כמה סוגים ואופציות של שדות טקסט לפי גודל ויעוד שדה הטקסט.
6. כפתורים/לחצנים למיניהם.
7. מדיה: רכיב תמונה ImageView וכפתור תמונה ImageButton, ורכיב וידאו VideoView.
8. סידור ממשק המשתמש ע"י תבניות.

קיימות מספר אופציות לסידור ממשק המשתמש אותן מספקת אנדרואיד ע"י מחלקות שמגדירות את סוגי התבניות:

i. `AbsoluteLayout`

ii. `FrameLayout`

iii. `LinearLayout`

iv. `RelativeLayout`

v. `TableLayout`

9. מיקום הפריסה, גודל, ריפוד ושוליים.

10. שמירת מצב הפעילות.

מעבר מפעילות אחת לפעילות אחרת גורם למעבר ממצב למצב, לפעמים יש צורך בשמירת מצב

הפעילות הנעלמת מהמסך. היכרות עם כלים ופעולות שעוזרים בשמירת המצב:

`SaveEnabled()`, `SetSaveEnabled()`, `OnSaveInstanceState()`

11. טיפול בשינוי תצורה.

חלק מהמכשירים עוברים שינוי תצורה בזמן ריצת פעילות מסוימת, כגון שינוי כוון המסך או

הצגת/הסתרת מקלדת, שינוי שפה וכו'. תגובת המערכת בדרך כלל לשינויים כאלה ע"י השמדה

ויצירה מחדש של הפעילות.

i. זימון `OnDestroy()` ומיד זימון `OnCreate()`.

ii. מימוש הפעולות `OnRestoreInstanceState()`, `OnSaveInstanceState()`.

דרכי הוראה

חשוב להתנסות עם התלמידים בכתיבת ממשק פשוט בשתי השיטות. השיטה התכנותית (בשפת סי

שארפ או ג'אווה) והשיטה ההצהרתית בתבנית XML, תוך דגשים על ההבדל בין שתי האופציות.

(בפרט כאשר רוצים לבצע סידור לרכיבים ומתן ערכים לתכונות של כל רכיב) ועל היתרון של כתיבה

בתבנית XML בשני מקרים אלו.

חשוב להסביר מדוע אי אפשר לוותר על הקוד (בסי שארפ או ג'אווה) לצורך טעינת והצגת התבנית

וטיפול באירועים ובתכני רכיבי ה UI האמורים להשתנות כתוצאה מהאינטראקציה עם המשתמש.

לצורך התנסות מעמיקה חשוב לבצע דוגמה מעשית שכוללת בניית ממשק בעזרת תבנית XML

שכוללת לפחות שדה קלט, שדה טקסט לדוגמה, וכפתור. אפשר לקחת דוגמה לחישוב ה BMI או

מחשבון המרת מטבעות שכוללת שני שדות טקסט לקלט (הגובה והמשקל או הערך להמרה ושער

המטבע), כפתור לצורך ביצוע החישוב וטקסט להצגת התוצאה. אפשר לשדרג את הדוגמה ע"

הוספת תמונות המתאימות לתוצאות או צבעים וכו'. אפשר לתת לתלמידים לבחור את הדוגמה כך

שתהיה לקוחה מתחומי העניין שלהם.

חשוב שהתלמיד יתנסה בבניית אותו ממשק בסידורים שונים (תבניות שונות).

יש לתרגל עם התלמידים איתור רכיב גרפי באמצעות הפלטה (Palette) של רכיבי ה UI שמספק סביבת העבודה (eclipse, visual studio וכו') בזמן עריכת תבנית XML. ומשם לאתר את התיעוד והממשק או לחפש את ממשק הרכיבים באתר הרשמי של אנדרואיד.

<http://developer.android.com>.

כדי לזהות שמירת מצב חשוב לבצע תרגיל של פעילות שמכילה לפחות שדה טקסט אחד, במהלך הניסוי תתבצע כתיבה לתוך השדה וביצוע שינוי תצורה כמו סיבוב מסך.

דרכי הערכה אפשריות לפרק זה:

1. בחינה מעשית או עבודת בית עם הגנה בכיתה הדורשת בניית ממשק משתמש שהמורה מציג.
2. הוספת שורות קוד המאפשרות שמירת הנתונים שכבר הוכנסו לממשק המשתמש בזמן שינוי מצב המסך מאנכי לאופקי.

חלוקת שעות:

נושא	עיוני	התנסות
תבניות XML שימוש במשאב תבנית XML מקוד (סי-שארפ או ג'אווה) בהתאם לבחירה, ורכיבי ממשק.	5	10
סידור הממשק ע"י תבניות.	4	8
שמירת מצב הפעילות בשינוי תצורה.	1	2

פרק 7 – תכנות מונחה אירועים ושימוש במחלקה הפנימית

הקדמה

בפרק זה נציב את מנגנון האינטראקציה בין המשתמש ליישום. האינטראקציה מתבצעת דרך מנגנון האירועים, מערכת האנדרואיד מדווחת ליישומים ולרכיבים הגרפיים על אירועים שמתרחשים והקשורים לאותו רכיב.

מטרות כלליות

טיפול באירוע לחיצה ע"י XML, היכרות ומימוש אירועים ומאזינים.

מטרות ביצועיות:

1. התלמיד יתכנת תגובה לאירוע לחיצה בשיטת XML.
2. התלמיד יסביר את תהליך הטיפול באירוע: אירוע ← מאזין ← תגובה.
3. התלמיד יתכנת תגובה לאירועים המטופלים ע"י מאזינים (2 מאזינים לפחות). לפי השלבים: יצירת המאזין ← רישום/הצמדת המאזין לרכיב ← כתיבת התגובה.
4. התלמיד ייצור אובייקט מאזין: ה Activity מממשת מאזין, מחלקה פנימית כמאזין, יצירת עצם מאזין בשורה (Inline).

מושגים והכוונה:

1. אירועים Events :
כאשר מתרחש אירוע בממשק המשתמש, הפריט (View) הקשור לאירוע הוא האחראי ללכוד ולטפל באותו אירוע, מערכת האנדרואיד והרכיבים בנויים לכך.
2. מאזינים:
הינם ממשקים המהווים חלק מה View, כדי לבצע טיפול באירוע צריך לממש את המאזין המתאים ודריסת שיטת התגובה מהם:
i. View.OnClickListener
ii. View.OnLongClickListener
iii. View.OnFocusChangeListener
iv. View.OnKeyListener
v. View.OnTouchListener
vi. View.OnCreateContextMenuListener
3. מחלקות פנימיות (Inner Classes) והרשאות גישה.
4. מימוש מאזין כמחלקה פנימית.

5. (בהתאם לשפה - שימוש במאזין באמצעות delegate)

דרכי הוראה:

להמשיך את הדוגמאות והתרגילים הקודמים מפרק ממשק משתמש גרפי ולהוסיף אירועים שונים בשיטות שונות.

דרכי הערכה אפשריות לפרק זה:

1. מטלה ביצועית במעבדה או בבית לפעילות עם ממשק משתמש שמגיב לאירועים: כמו מחשבון, הגרלות, אלבום תמונות וכו'.

חלוקת שעות:

נושא	עיוני	התנסות
טיפול באירוע לחיצה בשיטת XML	1	1
בניית מאזינים: פעילות מממשק, מחלקה פנימית ויצירת מאזין בשורה.	2	5

פרק 8 – מסרים Intents

הקדמה

הרכיבים העיקריים של האפליקציה services, activities, ו-broadcast receivers מופעלים ע"י מסרים שנקראים intents. ה intent נחשב לכלי קשירה בזמן ריצה בין רכיבים אלו ששייכים לאותו יישום או ליישומים שונים, ומהווה מנגנון משלוח מסרים/ נתונים.

משימושיו הנפוצים, מעבר מפעילות (מסך) לפעילות אחרת, הפעלת שירותים services ועוד אחרים הקשורים לרכיבים שנלמד בהמשך ואז יפורט וידגש השימוש ב intent. לדוגמה פעילות יכולה לבקש ע"י מסר להפעיל פעילות אחרת.

מטרות כלליות:

היכרות עם תפקיד העצם Intent סוגים: Implicit, Explicit. ומסנני מסרים - Intent filters. פעילות המחזירה תוצאה. Activity for Result.

מטרות ביצועיות:

1. התלמיד יסביר את משמעות המסרים ותפקידיהם.
2. התלמיד יסביר משמעות כל רכיב מרכיבי המסר.
3. התלמיד יפעיל בתכנות שני סוגי המסרים ויבחין ביניהם.
4. התלמיד ינווט ממפעילות לפעילות אחרת באותו יישום בעזרת intents (שני הסוגים).
5. התלמיד ישתמש במסנני המסרים. שימוש במסנני מסרים קיימים ויצירת מסננים מסרים משלו.
6. התלמיד יזמן פעילות מחוץ ליישום כמו דפדפן, בוחר תמונות, בוחר איש קשר וכו'.
7. התלמיד יתכנת העברת נתונים בין פעילויות.
8. התלמיד יתכנת קבלת תוצאה מפעילות קיימת (לדוגמה בוחר תמונות).
9. התלמיד יבנה פעילות שמחזירה תוצאה, ובניית פעילות המפעילה את הפעילות ומטפלת בתוצאה המוחזרת (ActivityResult).
10. התלמיד יסביר את מחסנית המשימות ואת התהליך המתבצע על כל פעילות בזמן הצגה או העלמה מהמסך (אין צורך בשינוי או השפעה על מצב מחסנית המשימות).

מושגים והכוונה:

1. העצם Intent:

עצם משייג מבנה נתונים מופשט, ונראה כתבילה של מידע שמעניינת את מקבל המסר, ובנוסף

מידע שמעניין את מערכת האנדרואיד. כמו שם הרכיב, Component name שמטפל ב Intent, הפעולה Action ועוד.

i. מרכיבי ה Intent:

I. Category

II. Data

III. Action Name

IV. Component Name

V. Flags

VI. Extra

ii. Explicit Intent או מסר מפורש,

בסוג זה מציינים במפורש את הרכיב שאמור להיקרא על ידי מערכת האנדרואיד, באמצעות המחלקה (ב סי-שארפ או ב ג'אווה) כמזהה.

אפשר להוסיף נתונים למסר ע"י הפעולה PutExtra(key,value);

זימון פעילות ע"י מסר מפורש, מתבצע ע"י זימון פעולות ההקשר המזמין (פעילות, שירות

וכו'). StartActivity(Intent(, StartActivityForResult(Intent(, ו StartActivity(Intent(.

iii. Implicit Intent או מסר מרומז,

בסוג זה לא מציינים את המחלקה (ב סי-שארפ או ב ג'אווה) שאמורה להיקרא, אלא מציינים

את הפעולה שאמורה להתבצע. (ייתכן עוד URI להפעולה).

בהעברת וקבלת נתונים ב Implicit Intent הרכיב שמקבל את המסר יכול לקבל את הנתונים

מהמסר ע"י פעולות אחזור (getters).

לדוגמה: GetData(), GetAction(), GetExtras() () ועוד.

iv. מסנני המסרים Intent filters: תפקידם ליידע את המערכת באילו מסרים מרומזים צריך

לטפל. כל מסנן מתאר יכולת של הרכיב, איזו קבוצת מסרים הרכיב מוכן לקבל.

v. מחסנית המשימות.

vi. פעילות מחזירה תוצאה (Activity for result).

דרכי הוראה:

ליצור שתי פעילויות ולבצע ניווט הדדי תוך שימוש במסרים והעברת נתונים וקריאתם כדי להמחיש את אחד מתפקידי המסרים.

ליצור דוגמאות בהן המסרים מזמינים פעילויות מחוץ ליישום.

לבצע דוגמה המשמשת מסנני מסרים קיימים ו/או יצירת מסנני מסר ובצוע שימוש בהם.

לבצע הדגמה להעברת נתונים ע"י מסרים.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע מסכמת שמבצע אותה התלמיד בכיתה או בבית, לדוגמה:
יישום שמכיל שתי פעילויות, הפעילות הראשונה היא ממשק של מספר שמייצג סכום כסף ושני שדות ומציינים ממה למה לבצע המרה.
הפעילות השנייה מופעלת למטרת תוצאה שמקבלת את הערכים מהפעילות הראשונה מבצעת המרה ומחזירה את התוצאה לפעילות המבקשת.
במעבר מפעילות הראשונה לשנה אפשר להדגים שני סוגי המסרים.

חלוקת שעות:

נושא	עיוני	התנסות
משמעות המסרים וסוגיהם וביצוע הדגמה לכל סוג	1	2
מסנני מסרים	1	2
פעילות למטרת תוצאה	2	4

פרק 9 – רכיבים מתקדמים: תפריט, תיבת דו שיח, הודעות, ה Status bar וטיפול באירועים

הקדמה

בפרק זה נעסוק ברכיבים מתקדמים, מתי ולמה רצוי להוסיף אותם ודגשים נוספים. חשוב לציין שבפרק זה ישנם הבדלי תכנים בהתאם לשפה שנבחרה. אפשר להציג לתלמידים את האפשרויות שיש בשפה שבחרת אל מול אלו של השפה שלא נבחרה.

מטרות כלליות:

היכרות עם תפריטים, תיבות דו שיח, הודעות טואסט (Toasts) ושורת מצב.

מטרות ביצועיות:

1. התלמיד ייצור תפריטים לסוגיהם.
2. התלמיד ייצור תיבות דו שיח לסוגיהן.
3. התלמיד ייצור הודעות טואסט Toast Notifications.
4. התלמיד ייצור הודעות שורת מצב ומימוש התגובה הקשורה בבחירת ההודעה.

מושגים והכוונה:

(בפרק זה ישנם הבדלי תכנים בהתאם לשפה שנבחרה – רצוי להדגיש הבדלים אלו)

1. תפריטים Menus:

- i. Options menu
- ii. Context menu
- iii. טיפול באירועי התפריטים: דריסת onOptionsItemSelected, ודריסת onSelectContextItem של Activity.

2. תיבות דו שיח Dialogs:

- i. Custom Dialog
- ii. ProgressDialog
- iii. AlertDialog
- iv. TimePickerDialog
- v. DatePickerDialog

3. טיפול באירועי תיבות הדו-שיח: (בהתאם לשפה שנבחרה)

- i. באמצעות delegate (סי-שארפ).
- ii. שימוש ב:

- I. DialogInterface.OnClickListener
- II. DialogInterface.OnMultiChoiceClickListener
- III. OnTimeSetListener
- IV. OnDateSetListener

4. הודעות Notifications:

- i. הודעות טואסט Toast Notifications
 - ii. הודעות שורת מצב Status Bar Notifications
5. טיפול באירוע בחירת הודעת שורת המצב: SetLatestEventInfo.

דרכי הוראה:

בתפריטים: ליצור פעילות המדגימה את שני סוגי התפריטים.
 בתיבות דו שיח: ליצור פעילות המדגימה את כל סוגי תיבות הדו שיח שנלמדו. ומימוש הטיפול באירועים הקשורים בתיבת הדו-שיח.
 בהודעות שורת המצב: ליצור ממשק של שני כפתורים אחד מציג את ההודעה והשני מבטל. וליצור תגובה לבחירת ההודעה שמעבירה את המשתמש למסך אחר.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע במעבדה או בבית המדגימה את כל מה שנלמד. לנסות לשלב את החידושים בדוגמאות ופרויקטים שנוצרו בפרקים קודמים (פרק ממשק משתמש גרפי ופרק טיפול באירועים).
 לדוגמה: תוספת תפריטים ותיבות דו שיח ל BMI , למחשבון או לממיר המטבעות.

חלוקת שעות:

נושא	עיוני	התנסות
תפריטים	1	2
תיבות דו שיח	2	4
הודעות	2	4

פרק 10 – משאבי האפליקציה Application Resources.

הקדמה

משאבי האפליקציה הם צרכים חיצוניים שנצטרך להם ביישום שלנו, כגון תמונות, מחרוזות, צבעים, סגנונות ועוד. יש כאן הרחבה של תכנים שהוצגו בפרק 5. כאמור משאבים אלו אמורים להימצא בתיקה המתאימה תחת תיקיית המשאבים בפרויקט "res". שיטה זו מאפשרת למפתחים להתאים את היישום המפותח לפי סוג המכשיר, שפה ומצב מסך המכשיר (לפני או אחרי סיבוב).

מטרות כלליות:

היכרות עם סוגי המשאבים, ארגוןם בפרויקט תחת "res", שימוש בהם (במחלקה R), יצירת סוגים שונים מהמשאבים.

מטרות ביצועיות:

1. התלמיד יסביר את המבנה של תיקיית המשאבים "res".
2. התלמיד יסביר את הקשר בין המשאבים ואת המחלקה Resource שנוצרת באופן אוטומטי.
3. התלמיד יצור ערכי טקסט כמשאב XML, ואת הזיקה עם תבניות למספר שפות.
4. התלמיד ייצור תפריט, צבעים, מערכים, ממדים dimensions.
5. התלמיד ייצור הנפשה ב XML.

מושגים והכוונה:

1. אספקת המשאבים Providing Resources: סוגי המשאבים, איפה לשמור אותם, התאמת המשאבים לתצורות המכשיר.
2. גישה למשאבים Accessing Resources: איך לגשת למשאבים ולהשתמש בהם, אם זה מקוד (סי-ארפ או ג'אווה) או ממשאב XML אחר.
3. טיפול בשינויים בזמן ריצה Handling Runtime Changes: איך לנהל ולהשתמש במשאבים כאשר מתרחש שינוי תצורה בזמן שיישום נמצא בפעולה.
4. תבניות שפות Localization: הצורך בהתאמת היישום ליותר משפה.
5. סוגי משאבים Resource Types: איך לכתוב ולהשתמש בסוגים השונים של המשאבים ע"י רכיבי XML. מהסוגים הקיימים:
 - i. מחרוזת String,
 - ii. צבעי מצב Color State List,

- .iii רכיבים הניתנים לציור (תמונות או XML וכו') Drawable,
- .iv פריסה Layout,
- .v תפריט Menu,
- .vi סגנון Style,
- .vii הנפשה Animation,
- .viii סוגים אחרים ידועים כמו Booleans, arrays, integers, dimensions, colors.

דרכי הוראה:

פרק זה מרכז את סוגי המשאבים שהתלמידים הכירו בעבר (או שילמדו בהמשך). לכן אפשר ללמד פרק זה באופן נפרד או כתוספים לפרקים המתאימים לסוג המשאב. לדוגמה כשיוצרים פעילות אז להרגיל את התלמידים בשימוש בערכים של משאב המחרוזות String לאגן את ערכי הטקסט החזותיים המשומשים במפעילות, כשלומדים על התפריטים אז נלמד את המשאב של התפריטים, כשלומדים יישומונים (App Widgets) מלמדים את המשאב של ממדים dimensions, בפרק ההנפשות (אנימציות) מלמדים את משאב ההנפשה וכו'. שאר הנושאים ניתן ללמד אותם באופן נפרד.

דרכי הערכה אפשריות לפרק זה:

1. בחינה עיונית שמבקשת תשובות והסבר לכל סוגי המשאבים.
2. בחינת מעבדה שבה התלמיד מתבקש לצור מספר משאבים מסוגים שונים תוך שיוך המשאבים לתיקה המתאימה בתיקיית המשאבים " /res".

חלוקת שעות:

נושא	עיוני	התנסות
מבנה תיקיית המשאבים וסוגיהם	2	2
יצירת וארגון סוגים שונים מהמשאבים	1	4

הקדמה

מערכת האנדרואיד מספקת מספר אפשרויות לאחסון נתונים, בחירת אופציית האחסון צריכה להיעשות בהתאם לדרישות של מפתח היישום ובצרכים של היישום. כגון כמות הנתונים הנשמרים, האם הנתונים הנשמרים פרטיים ליישום או משותפים עם יישומים אחרים וכו'.

מטרות כלליות:

שמירת העדפות/הגדרות, אחסון פנימי, אחסון חיצוני SQLite Database.

מטרות ביצועיות:

1. התלמיד יבנה ממשק משתמש לשמירת הגדרות/העדפות Preferences.
2. התלמיד ישמור וישחזר קבצים בזיכרון המכשיר. וידע את מאפיינים שמירה זו.
3. התלמיד יבדוק קיום מדיה חיצונית במכשיר לשמירה נתונים בקבצים ויתנסה בשמירה ובאחיזור קבצים בזיכרון החיצוני. ויכיר וידע לתאר את מאפייני שמירה זו.
4. התלמיד יבנה טבלה במסד הנתונים SQLite Database וידע לבצע: שליפה, הוספה, עדכון ומחיקה. כשזה מתאפשר בשפת SQL או במחלקות המקלות על השימוש במסד נתונים זה.
5. התלמיד יציג תוכן מסד נתונים ע"י ListView, בשימוש במתאם SimpleCursorAdapter.
6. תלמיד יטפל תכנותית באירוע לחיצה רגילה ו/או ארוכה על פריט ברשימה ListView.

מושגים והכוונה:

1. אחסון העדפות משותפות Shared Preferences. ליישום בזיכרון המכשיר ע"י יצירת ממשק משתמש מסך העדפות PreferenceActivity
2. אחסון העדפות משותפות באופן ידני ע"י ממשק כלשהו.
3. אחסון פנימי Internal Storage (זיכרון המכשיר) קריאה/כתיבה של קבצים בזיכרון הפנימי.
4. קריאת קובץ שנמצא במשאבי היישום Resource File.
5. אחסון חיצוני External Storage, כתיבה\ קריאה בהתקן אחסון חיצוני (לדוגמה SDCARD).
6. השימוש בהרשאות: android.permission.WRITE_EXTERNAL_STORAGE.
7. אחסון בבסיסי נתונים SQLite Databases:

מסד הנתונים SQLite הוא מסד נתונים קוד פתוח אשר מוטבע באנדרואיד. הוא מאפשר אחסון וניהול נתונים בעזרת שאלות SQL שמספקת SQLite. המחלקות השימושיות בו הן:

- i. SQLiteDatabase,
- ii. SQLiteOpenHelper ,

- ContentValues. .iii
- Cursor, .iv
- 8. הצגת נתונים בעזרת רשימה ListView:
- i. SimpleCursorAdapter
- ii. פעילות רשימה: ListActivity
- 9. טיפול באירועי לחיצה על פריט ברשימה:

דרכי הוראה:

יש לסקור בפני התלמידים את תכונות כל שמירה שכולל רמת הפרטיות והאבטחה. את הצורך לשימוש בכל שיטה.

בהתחלה יש לבצע דוגמה של שמירת שני נתונים "משתמש" ו"סיסמה". שמירת הנתונים תבצע בכל השיטות כך התלמידים יחשפו לדרכי השימוש בכל שיטה.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע בה מתבקשים התלמידים ליצור ממשק משתמש לטיפול במסד נתונים כולל שמירה, שליפה, מחיקה ועדכון. והצגת הנתונים ע"י רשימה ListView.

חלוקת שעות:

נושא	עיוני	התנסות
אחסון העדפות משותפות Shared Preferences	1	2
אחסון פנימי	1	2
אחסון חיצוני	1	2
SQLite Database	7	14
SimpleCursorAdapter , ListView	2	4

פרק 12 – תכנון וניתוח אפליקציה, שימוש ב UML, תיאור מסכים, תרשים זרימה

הקדמה

בפרק זה התוודע לשיטת מידול בעזרת UML (Unified Modeling Language). זו שפת מידול, בעלת אופי ויזואלי, המאפשרת תיאור מופשט של מערכת. היא ממדלת מחלקות, רכיבים, רצפים, קשרים ועוד, באופן גרפי על ידי תרשימים.

מטרות כלליות:

אפיון וניתוח יישום, ניתוח מחלקות ע"י תרשימי UML, תרשים זרימה ותיאור מסכים, זיהוי מרכיבים ייחודיים, תיאור הנתונים הנשמרים ואופן יצירתם

מטרות ביצועיות:

1. התלמיד יגדיר את מטרת היישום.
2. התלמיד יאפיין עצמים ומרכיבים חיוניים ביישום.
3. התלמיד יתאר את המראה הגרפי של היישום.
4. התלמיד יתאר את תרשים הזרימה של המסכים.
5. התלמיד יקשר בין המסכים ובין הרכיבים והמשימות הייחודיות של היישום.
6. התלמיד יתאר ב UML את המחלקות החיוניות והקשרים ביניהם.
7. התלמיד יראה כיצד לנהל לוח זמנים וסידור בניית היישום בשלבים.
8. התלמיד ימדוד את התקדמותו בפיתוח היישום בהתאם לניתוח שביצע.

מושגים והכוונה:

1. אפיון האפליקציה.
2. תיאור הצורך והשימוש באפליקציה.
3. זיהוי משימות ראשיות באפליקציה (פעילויות, שירותים, מקלטים, מסרים, ספקי תוכן).
4. שרטוט סקיצות למשימות.
5. תכנון תרשים זרימה לאפליקציה.
6. יצירת תבניות או מסכים.
7. קביעת הנתונים שרוצים לשמור ו/או לבצע גיבוי, מבנה ושיטת השמירה לצורך זה.
8. ניתוח מחלקות ע"י תרשים UML.

דרכי הוראה:

הצגת יישום מיצג וביצוע כל האפיונים, ניתוחים, התרשימים, הצרכים, המסכים וכו'. הניתוח שיעשה ביישום זה יהיה ברמת ביצוע שתתאים ברמתה לנדרש בפרויקט הגמר. (גם אם לא באותו היקף)

דרכי הערכה אפשריות לפרק זה:

1. בשלב זה התלמיד יבחר לעצמו יישום שירצה אולי לפתח ויבצע את ניתוח והאפיון המתאימים. ברמת ביצוע שתתאים ברמתה לנדרש בפרויקט הגמר.

חלוקת שעות:

נושא	עיוני	התנסות
מטרת היישום והצרכים	1	2
עצמים ורכיבים חיוניים ליישום	1	2
שרטוט הישויות ביישום והקשרים ביניהם כולל מסכים, שירותים, מקלטים, ספקי תוכן, נתונים הנשמרים.	2	4

פרק 13 – ספקי תוכן Content Providers

הקדמה

מאחר שאין שטח אחסון משותף לכל האפליקציות באנדרואיד, ספקי התוכן היא סוג מיוחד של מאגר הנתונים שחושף דרכים סטנדרטיים כדי לאחזר ולטפל נתונים המאוחסנים. אנדרואיד מגיע עם מספר ספקי תוכן שימושיים.

מטרות כלליות:

היכרות עם ספקי התוכן, מטרתם והשימוש בספקי התוכן הקיימים של אנדרואיד, בהתאם ביצוע שאליות על ספק התוכן.

מטרות ביצועיות:

1. התלמיד ישתמש בלפחות באחד מספקי התוכן הקיימים: לאנשי קשר, יומן השיחות, תכני דפדפן, הגדרות ההתקן, קבצי מדיה, יומן.
2. התלמיד יבצע לפחות שאלית אחזור query().

מושגים והכוונה:

1. משמעות ספקי התוכן והסוגים אותם מספקת המערכת ע"י השימוש ב URI המתאים.
2. שימוש בספק נתונים, כגון אנשי קשר, יומן השיחות, תכני דפדפן, הגדרות ההתקן, קבצי מדיה, יומן.
3. ביצוע שאליות על ספק תוכן: אחזור, הוספה, עדכון ומחיקת נתונים.
(onCreate(), getType(), delete(), update(), insert(), query())

דרכי הוראה:

יש להבהיר את משמעות ספקי התוכן וחשיבותם, ליצור טבלה שמכילה את ה URI המתאים לכל אחד מסוגי הספקים. ולבצע שימוש בספק נתונים אחד לפחות עם התלמידים תוך שימוש בכל סוגי השאליות האפשריות.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע בה התלמיד יתבקש להשתמש בספק תוכן אחר ממה שהודגם בכיתה, ויקבל הנחיות חיוניות לביצוע השאליות, שעליו לבצע.

חלוקת שעות:

נושא	עיוני	התנסות
משמעות ספקי התוכן וה URI.	2	1
שימוש בספקי תוכן.	2	7
ביצוע שאילתות	4	8

פרק 14 – תהליכים ותהליכונים Threads & Processes

הקדמה

תהליכון או Thread הוא רצף של פעולות שמתבצעות באופן עצמאי במטרה לבצע משימה מסוימת, וזאת במקביל לתהליכונים אחרים.

במערכת האנדרואיד מוקצה תהליך יחיד ותהליכון עבור כל יישום שמבקש לרוץ. פרק זה דן איך התהליכים Processes והתהליכונים Threads ממומשים ועובדים ביישומי האנדרואיד.

מטרות כלליות:

היכרות עם הרעיון של עבודה של תהליכים במקביל בשונה מהתכנות הסדרתי שנלמד במקצוע המוביל. ומימוש באפליקציה.

מטרות ביצועיות:

1. התלמיד יסביר את משמעות התהליכים והתהליכונים באופן כללי ותיאורטי.
2. התלמיד יסביר באופן תיאורטי את צורך המערכת בסיום תהליכים ואיך עושה זאת מערכת האנדרואיד לפי עדיפויות.
3. התלמיד יבנה תהליכון ע"י Thread.
4. התלמיד יעדכן את ממשק המשתמש ע"י Thread אחר בעזרת Handler או ע"י המשימה האסינכרונית AsyncTask.

מושגים והכוונה:

1. תכנות מקבילי: עיקרון מימוש המקביליות בעולם המחשב והצורך בו.
ה- Thread כדוגמה למימוש תכונה זו בעולם המחשבים.
תכונות ומרכיבי התהליכון (ב ס' שארפ / ג'אוה)
2. יצירת תהליכון Thread. (בהתאם לשפת העבודה! שימוש ב-runnable בג'אוה)
3. תהליכונים מרובים: תזמון בין רצפי פעולה - threads scheduling, סיום חייו של thread, המתודה isAlive ו-join.
4. שמירת שלמות הנתונים. Synchronization ובעיית ה-Deadlock.
5. מתודות wait() ו-notify().
6. תהליכים Processes: כל רכיבי אותו יישום רצים על אותו תהליך (Process) שמקצה מערכת הלינוקס לכל יישום.

7. מעגל חיי התהליך: מערכת האנדרואיד מנסה לתת לתהליך לרוץ זמן ארוך ככל האפשר, בסופו של דבר צריך לסיים חלק מהתהליכים לצורך התייעלות בניצול וניהול הזיכרון. בחירת התהליך המיועד לסיום מסתמכת על רמת עדיפות של 5 רמות:

- i. Foreground process
- ii. Visible process
- iii. Service process
- iv. Background process
- v. Empty process

8. תהליכונים Threads והיישומים באנדרואיד.

המערכת מקצה לפחות תהליכון אחד עבור כל יישום המבקש לרוץ. תהליכון זה נקרא ה"תהליכון הראשי" - "main thread". תהליכון זה אחראי על שיגור אירועי ממשק המשתמש הגרפי.

הוא גם התהליכון שבו מתבצעת האינטראקציה בין היישום לרכיבי ערכת הכלים של ממשק המשתמש הגרפי של אנדרואיד (Android UI toolkit). לכן תהליכון זה נקרא גם "תהליכון ממשק משתמש" או "UI thread".

9. התהליכונים הפועלים (Worker threads): תהליכון לביצוע משימה מקבילה לתהליכון הראשי בלי לחסום את ריצתו או לשבש את ריצתו. תהליכונים אלו נעזרים בדרך כלל בפעולות הבאות: (יתכנו הבדלים במימוש בין סי-שארפ לג'אווה)

- i. Activity.runOnUiThread(Runnable)
- ii. View.post(Runnable)
- iii. View.postDelayed(Runnable, long)

10. שינוי ממשק המשתמש מתוך Thread ע"י Handler בשיטת שליחת ההודעות.

11. שינוי ממשק המשתמש מתוך Thread ע"י משימה אסינכרונית AsyncTask: מאפשרת לבצע משימה אסינכרונית על ממשק המשתמש. כך שמבצעת את הפעולות החוסמות ב"תהליכון הפועל" ואז מעדכנת את התוצאות על תהליכון ממשק המשתמש "UI thread", בלי לטפל בתהליכונים.

דרכי הוראה:

להסביר באופן פרונטאלי משמעות התכנות המקבילי ואת הגדרת התהליכון.

לבנות דוגמה פשוטה שמדגימה שני תהליכונים. למשל לבנות מחלקה שמרחיבה Thread שמכילה תכונה אחת של מחרוזת שערכה מאותחל בשיטה הבונה, לבצע קטע קוד מקבילי שמדפיס את תוכן התכונה. אחרי שמסיימים בונים שני עצמים עם שני ערכים שונים ולבצע הפעלה של שני התהליכונים ולצפות בתוצאות.

לחזור על הדוגמה בעזרת מימוש המחלקה Runnable (ג'אווה).

כדי להמחיש את הצורך ב Handler ו AsyncTask, לבנות פעילות עם Thread שמשנה משהו בתוכן ה UI (למשל מציג מספר אקראי בעזרת TextView או Toast). אז נדגיש את הבעיה שזה לא מתאפשר, לכן משתמש בHandler או AsyncTask כפתרון לבעיה.

דרכי הערכה אפשריות לפרק זה:

1. בחינה עיונית על משמעות התכנות המקבילי, כולל כתיבת קוד.
2. בחינת מעבדה או מטלת ביצוע לכתיבת קטע קוד שירויץ במקביל ולתת לתלמידים לבחור את השיטה שנוחה להם.

חלוקת שעות:

נושא	עיוני	התנסות
התכנות המקבילי Thread	2	4
Runnable (ג'אוה)	3	6
תהליכים Processes		2
Handler או AsyncTask (ללומדים ב סי-שארפ, להוסיף 9 שעות לנושא זה)	8	20

פרק 15 – שירות Service

הקדמה

ישנם שני סוגי "שירות" כל אחד בעל יעוד אחר, הראשון "started" הרגיל והשני "bound" שקשור לרכיב אפליקציה אחר (כגון פעילות אחרת). השירות מתחיל כ "started" אחרי זימון רכיב אחר את הפעולה StartService(), ומסתיים אחרי שהוא מסיים את עצמו ע"י הפעולה StopSelf() או שרכיב אחר מסיים אותו ע"י הפעולה StopService(). השירות מתחיל כ "bound" אחרי זימון רכיב אחר (הלקוח) את הפעולה BindService(), ומסתיים בזימון הלקוח את הפעולה UnbindService(). בפרק זה נתמקד רק בסוג הראשון started.

מטרות כלליות:

היכרות והבנת משמעות ותפקיד השירות, בניית שירות started, הבנה מהו מעגל החיים, הבנת ה IntentService, ושירותים קיימים המערכת.

מטרות ביצועיות:

1. התלמיד יסביר את משמעות השירות ואת הקיום של שני סוגים הרגיל, "started" והשני, "bound" שקשור לרכיב אפליקציה אחר.
2. התלמיד יבנה את השירות "started".
3. התלמיד יפעיל ויעצור שירות.
4. התלמיד יסביר את מעגל חיי השירות ואת היכולת לדרוס את הפעולות שבמעבר ממצב למצב לפי הצורך.
5. התלמיד יתכנת שירות חזית.
6. התלמיד יבצע תזמון לתחילת שרות בעזרת AlarmManager ולבצע שירות מחזורי כל פרק זמן.

מושגים והכוונה:

1. מעגל חיי השרות:
 - i. onCreate()
 - ii. onStartCommand()
 - iii. onBind()
 - iv. onUnbind()
2. מתודות/פעולות מעגל חיי השירות: במעבר ממצב למצב מופעלות הפעולות הבאות:
 - Starting, Running, Shut-down, Destroyed.

- v. OnUnbind()
- vi. OnRebind()
- vii. OnDestroy()

- כדי להתערב ולהשפיע במעבר ממצב למצב צריך לדרוס פעולות אלו (שהן חלק מהמחלקה Service). סדר הקריאה לפעולות שונה משני סוגי השירות "started" ו "bound".
3. בניית Started Service. כהרחבה של Service ו IntentService וההבדל ביניהם. (ה IntentService רץ ב thread נפרד מהפעילות, לדוגמה שירות שמוריד קובץ מהאינטנט).
 4. שירות חזית Foreground Services: שירות חזית הוא שירות שאמור להיות משהו ידוע למשתמש כפעיל. השירות הופך להיות שירות חזית אחרי זימונו לפעולה startForeground עם הודעת שורת המצב.
 5. השימוש AlarmManager לצורך תזמון תחילת שירות ותזמון שירות מחזורי.

דרכי הוראה:

רצוי להתחיל בהוראה פרונטאלית שמסבירים את משמעות השירות ותפקידו תוך כדי הדגמות של שעון מעורר ועדכון דוא"ל, עדכון מזג האוויר וכו'.

יש ליצור שירות מסוג "started", ולדרוס את פעולותיו, בתוך כל פעילות להציג את שם הפעילות בעזרת "התראת טואסט". לבנות פעילות עם 2 כפתורים כך שאחד מפעיל והשני עוצר את השירות מסוג "started", יש לבצע את תגובת כל כפתור לפי התפקיד שלו.

אפשר לשדרג את הדוגמה הקודמת לשירות חזית ע"י זימון השירות ל startForeground. בשלב המתקדם לבצע את אותה המשימה לתזמון תחילת השירות בעזרת AlarmManager ושימוש ב PendingIntent, ולשדרג אותה לגרסה נוספת החוזרת על הפעלת השירות כל פרק זמן קצוב. נושא השירות עוד יחזור בנושאים מתקדמים כגון תהליכונים (thread) ושירותי מערכת.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע – יצירת שירות והפעלתו.

חלוקת שעות:

נושא	עיוני	התנסות
סוגי השירות, ומעגל החיים .	2	2
בניית שירות כהרחבה של Service ו IntentService	3	7
תזמון שירות ושירות מחזורי	2	5
שירות חזית	1	2

מקלט שידורים (**Broadcast Receiver**) הוא מחלקה של היישום, שתפקידו להאזין למסרים (intends) שמשודרים. הוא מחלקה שמרחיבה את המחלקה BroadcastReceiver ורשומה ביישום AndroidManifest.xml. כל מקלט כזה נרשם לקליטת מסר או קבוצת מסרים, והינו מיועד ע"י מערכת האנדרואיד והיישום ששייך לו במסרים שמשודרים, כלומר מתפקד כמאזין למסרי היישום והמערכת לפי הרשאות ויעוד.

מטרות כלליות:

היכרות והבנת אופן הפעולה של מקלט שידורים BroadcastReceiver. בניית BroadcastReceiver, שימוש והפעלתו - שידור מסרים למקלט, קליטת שידורי מסרים משירותי המערכת.

מטרות ביצועיות:

1. התלמיד יסביר את מנגנון מקלט המסרים.
2. התלמיד יתכנת את תהליך הבנייה: בניית מקלט, רישום, שליחת מסרת קבלת המסר והתגובה.
3. התלמיד יסביר את הצורך בהרשאות אבטחה והשימוש בהן.
4. התלמיד ייצור הרשאה משלו למקלט שהוא בנה.
5. התלמיד ישתמש לכל הפחות במסר של שירות מערכת אחד כמו: סיום עליית המערכת, שיחה נכנסת, SMS, התקבל וכו'.

מושגים והכוונה:

1. מקלט השידורים או המסרים - BroadcastReceiver. מוגדר כמאזין למסרי היישום והמערכת לפי הרשאות ויעוד. לדוגמה המערכת משדרת ליישומים שסיימה העלאה לכן יופעל המקלט עם מסן android.intent.action.BOOT_COMPLETED, כמו כן לזיהוי שיחה נכנסת או SMS שהתקבל עם מסן android.provider.Telephony.SMS_RECEIVED וכדומה. ביצוע תגובה למסר הזה ע"י המתודה onReceive().
2. שידור מסרים Broadcast Intents על ידי void SendBroadcast(Intent intent).
3. קליטת מסרי מערכת האנדרואיד: המסרים המסופקים ע"י שירותי המערכת ו BroadcastReceiver.
4. הרשאות ואבטחה: שימוש בהרשאות קיימות והגדרת הרשאות אבטחה.

5. שימוש בהרשאות:
- יישום בסיסי אינו מכיל הרשאות לכל דבר במערכת, כדי לתת הרשאה מסוימת צריך לציין אותה בקובץ המניפסט של היישום ע"י המרכיב `uses-permission`. לדוגמה, היישום יקבל גם את המסרונים (SMS) הנכנסים, או לתת לו גישה למצלמה או ל GPS וכו'.
6. הגדרת דרישת הרשאות:
- אפשר להגדיר הרשאות לפי דרישות המתכנת בקובץ המניפסט ע"י הרכיב `<permission>`.
7. יישומון App Widgets:
- זהו יישום קטן, שמהווה תצוגה מצומצמת או זעירה של יישום מסוים שניתן להטמיע אותה באפליקציות אחרות כמו מסך הבית של מערכת האנדרואיד. יישומון זה יכול להתעדכן באופן תקופתי. לדוגמה: יישומון המציג מזג האוויר, יישומון המציג את סדר היום של לוח שנה וכו'.
- ליצירת יישום יש צורך ב `AppWidgetProviderInfo` שמספק תיאור ומאפיינים ליישומון, ו `AppWidgetProvider` שהינו סוג של `broadcast receivers` או מקלט מסרים, עצם כזה מקנה לאפליקציה יכולת קבלת מסרים (Intents) ששודרו ע"י המערכת או אפליקציה אחרת אפילו כאשר רכיבים אחרים של היישום אינם פעילים.

דרכי הוראה:

להבליט את חשיבות מנגנון זה תוך מתן דוגמאות ממסרי המערכת.

ליצור דוגמה של הפעלת שירות או הצגת פעילות כתגובה לקליטת מסר שנשלח ע"י יישום התלמיד או מסר של המערכת.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע - ליצור דוגמה של הפעלת שירות או הצגת פעילות כתגובה לקליטת מסר שנשלח ע"י יישום התלמיד או מסר של המערכת.

חלוקת שעות:

נושא	עיוני	התנסות
המחלקה <code>BroadcastReceiver</code> , בניית מקלט, רישום, שליחה וקליטת מסר.	3	6
שירותי מערכת, הרשאות ואבטחה.	2	4
App Widgets	3	6

פרק 17 – טלפוניה

הקדמה

פרק יעסוק בשימוש הקלאסי של הטלפון, שיחה והודעות טקסט. משמעות המילה "tele" ביוונית הוא רחוק (far), ומשמעות המילה "phone" ביוונית הוא קול או דיבור (sound/voice). (על בסיס האתר etymonline.com – אתר החוקר מקורות של מילים) בטלפונים הסלולריים התקשרות מאפשר גם שליחת מסרונים או בשם המקורי הודעות טקסט (SMS – Short Message Service) וגם הם הוכנסו לפרק זה.

מטרות כלליות:

היכרות עם אופן ברור מצב הטלפון, ביצוע שיחות חיוג, קבלת שיחה נכנסת, הודעות טקסט SMS שליחה וקבלת הודעה. אנדרואיד מספקת חבילה עם מחלקות שמטפלות בכל נושא הטלפוניה.

מטרות ביצועיות:

1. התלמיד יסביר כיצד לקבל את תכונות הטלפון.
2. התלמיד יבצע בקשת חיוג או ביצוע חיוג.
3. התלמיד יפעיל מאזין מצב הטלפון לזיהוי מצב הטלפון וזיהוי שיחה נכנסת.
4. התלמיד יזהה שיחה יוצאת בתכנות.
5. התלמיד ישלח ויקבל מסרון SMS.
6. התלמיד ישתמש במקלטי שידור BroadcastReceiver לצורך זיהוי שיחה יוצאת.
7. התלמיד ישתמש במקלטי שידור BroadcastReceiver לצורך זיהוי קבלת SMS.

מושגים והכוונה:

1. מנהל הטלפוניה TelephonyManager.
2. מאזין מצב הטלפון PhoneStateListener.
3. קבלת תכונות ומצב הטלפוניה של הטלפון.
4. שימוש במסרים (Intent) לבקשת חיוג עם Intent.ACTION_CALL. וביצוע חיוג ע"י Intent.ACTION_DIAL.
5. זיהוי שיחה נכנסת.
6. שליחת SMS ע"י Intent או ע"י מנהל המסרונים SmsManager.
7. הרשאות הנחוצות לחיוג, קבלת שיחה, שליחת SMS וקבלת SMS.

דרכי הוראה:

הדגמת השימוש במאזין מצב הטלפון PhoneStateListener ושימוש BroadcastReceiver. ההדגמה תכלול ביצוע שיחות, זיהוי שיחה יוצאת, זיהוי שיחה נכנסת, שליחת SMS וקבלתו. תרגיל מצוין יהיה ליישם שימוש בשירות קיים במערכת שבה מתנהל הטלפון. או תרגיל ל BroadcastReceiver ושימוש בהרשאות

דרכי הערכה אפשריות לפרק זה:

1. בחינה מעשית או מטלת ביצוע לסיכום הנושא על בסיס המטרות הביצועיות.

חלוקת שעות:

נושא	עיוני	התנסות
PhoneStateListener ו TelephonyManager	3	6
זיהוי שיחה יוצאת.	2	4
קבלת ושליחת SMS	3	6

פרק 18 – רכיבים מתקדמים: מולטימדיה, מצלמה, חיישנים, GPS, מפות.

הקדמה

רוב המכשירים המופעלים ע"י אנדרואיד מכילים רכיבי חישה ומקלט מערכת מיקום גלובלית. GPS. (Global Positioning System). מערכת אנדרואיד מספקת מרחב אפשרויות לנגינת קבצי קול, הפעלת וידאו, הצגת תמונות, הקלטת קול, ולכידת תמונה ווידאו. הפרק זה נלמד באופן בסיסי את השימוש ברכיבים אלו

מטרות כלליות:

שימוש במנהל החיישנים `SensorManager`. ובמאזין לחיישנים `SensorEventListener`, שימוש בחיישני תנועה, חיישני מיקום, חיישני סביבה, מיקום ומפות, נגינה אודיו וידאו, הקלטת קול, לכידת תמונה, צילום וידאו.

מטרות ביצועיות:

1. התלמיד יזהה את סוגי החיישנים האפשריים.
2. התלמיד ישתמש בחיישן אחד לפחות (מבין החיישנים וה-GPS), בשימוש במאזין החיישנים.
3. התלמיד יציג מפה ע"י `MapView`.
4. התלמיד יפעיל קבצי מדיה (קול, תמונה, ווידאו).
5. התלמד יסביר את קיום האפשרות ללכידת: קול, תמונה ווידאו, ואת המחלקות האחראיות על זה.

מושגים והכוונה:

1. מנהל החיישנים `SensorManager`.
 2. המאזין לחיישנים `SensorEventListener`.
 3. חיישני תנועה:
- משתמשים בהם לזיהוי תנועות המכשיר, כגון הטיה, ניעור, הנפה, סיבוב. חיישנים אלו מורכבים מ - מד תאוצה "accelerometer", ומד כוון "gyroscope". שניהם צריכים להיות חלק מהחומרה. ישנם עוד שלושה חיישני תנועה שיכולים להיות ממומשים בחומרה או בתוכנה. חיישן משיכה, חיישן תאוצה לינארית, וחיישני וקטור הסיבוב.
4. חיישני מיקום:
- מערכת האנדרואיד מספקת שני חיישנים שבנויים בחומרה כדי לאבחן את מיקום המכשיר, "חיישן השדה המגנטי של כדור הארץ" וחיישן כיוון.

5. חיישני סביבה:

מערכת האנדרואיד מספקת ארבעה סוגי חיישנים לחישה הסביבה : חיישן לחות, חיישן אור, חיישן לחץ אוויר וחיישן טמפרטורה.

6. מיקום ומפות:

ע"י אנטנת GPS מאפשרת מערכת האנדרואיד לקבל את הקואורדינטות העולמיות של מיקום המכשיר. המערכת מאפשרת גישה למפות שלה מה שמאפשר מראה ויזואלי אמיתי למיקום. LocationManager הינה המחלקה האחראית לשימוש וניהול כלי זה.

7. MediaPlayer: המחלקה האחראית להפעלת קבצי מוזיקה ווידאו.

8. AudioManager: מנהל מקורות וכלי השמעת קול.

9. MediaRecorder:

מאפשר הקלטת קול

מאפשר שליטה במצלמה וצילום תמונות או וידאו

דרכי הוראה:

לסקור את סוגי החיישנים האפשריים.

לתרגל עם התלמידים שימוש בחיישן אחד לפחות (מקבוצת החיישנים וה-GPS).

אם עובדים עם סימולטור שאינו מספק אופציות לכל החיישנים אבל מספק אופציה לשימוש ב-GPS (על ידי ה-DDMS) אז אפשר לתרגל רק את השימוש ב-GPS.

אם עובדים עם מכשיר אז עדיף לבצע עוד שימוש בחיישן, ההדגמה תהיה על המכשיר עצמו. חשוב לתרגל הפעלת קבצי קול ווידאו והפסקתם וגם על אופציות החזרה בהפעלת כל סוג.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע במעבדה או בבית לשימוש במקלט ה-GPS כמו חישובי מרחק, חישובי מהירות וכו'.
2. ניתן להוסיף הפעלת קול כתגובה לאירועים דוגמת לחיצה על כפתור.

חלוקת שעות:

נושא	עיוני	התנסות
סקירת סוגי החיישנים.	2	4
אופן השימוש בחיישנים ע"י מנהל החיישנים וההאזנה להם.	3	6
ה-GPS כדוגמה ומפות.	4	8
מדיה	3	6

הקדמה

בפרק זה נלמד לעצב רכיבי תצוגה (View) על ידי ציור דו-ממדי (2D), שנבצע באמצעות כלים העומדים לרשותנו במחלקה Canvas או יצירת רכיבים חדשים הניתנים לציור (Drawable). בנוסף נבצע אנימציה ואינטראקציה עם המשתמש.

מטרות כלליות:

ציור וגרפיקה בעזרת XML ו Canvas, ביצוע אנימציות ב XML ועל גבי SurfaceView.

מטרות ביצועיות:

1. התלמיד ישתמש בכלים הגרפיים העומדים לרשותו.
2. התלמיד יבצע ציור, צביעה ויצירת רכיבים גרפיים: או בשיטת XML או שימוש ב Canvas.
3. התלמיד יבצע סוג אחד של אנימציה מבין הסוגים שהוצגו: AnimationDrawable, ViewAnimation או על ידי ציור על View על ידי הרחבה של View או SurfaceView.

מושגים והכוונה:

1. ציור בעזרת XML.
2. ציור על View:
מתבצע ע"י דריסת הפעולה View.OnDraw(). משתמשים בשיטה זו כאשר הרכיבי לא צריך לבצע אנימציה מהירה אן כבידה, יותר מתאים לגרפיקה יותר סטטית וקבועה.
3. Canvas:
זאת המחלקה שמהווה את הממשק הוויזואלי של הרכיבי הגרפי. מחלקה זו מכילה את הפעולות האחראיות על ציור צורות גרפיות שונות.
4. Paint:
מחלקה המחזיקה במידע על צבע וסגנון כדי לתאר איך מצירים צורות גיאומטריות, טקסט ומפת סיביות Bitmap.
5. SurfaceView:
מחלקה המתאימה לביצוע אנימציות. הייחודיות שלה בזה שהיא מציעה תהליכון (Thread) עצמאי לביצוע הצביעה, כך שהרכיב לא ממתין בתורו עד שהמערכת מצירת אותו. SurfaceHolder לפתרון בעיית העדכון (שינוי) הגרפי על SurfaceView.

6. Drawables:

אנדרואיד מציעה חבילת מחלקות לציור בדו-ממדי `android.graphics.drawable`. בחבילה זו נמצאות `BitmapDrawable`, `PictureDrawable`, `ShapeDrawable`, `LayerDrawable` ואחרות. ברכיבים אלו אפשר להשתמש במשאבים קיימים כמו תמונות, וליצור אותם ע"י XML. חבילה זו מאפשרת אנימציה `frame-by-frame` שניתן לכתוב אותה בשיטה XML.

i. אנימציה בעזרת המחלקה `AnimationDrawable`.

ii. אנימציה בשיטת `View Animation`.

iii. אנימציה בעזרת תהליכון רגיל (`Thread`, `Timer`) על `SurfaceView` שבאחריות `SurfaceHolder`.

דרכי הוראה:

התלמידים יתרגלו את שיטות הצביעה והציור ב XML, באופן מעשי. להדגים אנימציה ע"י `AnimationDrawable`. ואנימציה בשיטת `Animation View` לבצע הרחבה למחלקה `View` ולבצע דריסה לפעולה `onDraw(Canvas canvas)`. להשתמש ב `Canvas` לבצע צביעה וציור, לדוגמה צביעת רקע, ציור צורות (קו, מלבן, מעגל וכו') תוך שימוש במחלקה `Paint`, ציור תמונת `Bitmap`. בהמשך לשנות את הירושה בדוגמה הקודמת מ `View` ל `SurfaceView`, ולזהות שלא התבצע ציור. וכאן בא הצורך והתפקיד של ה `SurfaceHolder`. לבצע אנימציה כתגובה לאירוע מגע במסך. לדוגמה הזזת תמונה או עיגול על המסך בהתאם לאירוע המגע. לבצע אנימציה שמשתמשת ביותר מרכיב אנימציה אחד על המסך. לדוגמה שני עיגולים או תמונות שזזים באופן שרירותי על המסך.

דרכי הערכה אפשריות לפרק זה:

1. מטלת ביצוע מודרכת ליצירת רכיב מעוצב. או ממשק עם רכיבים שמבצעים אנימציות בעזרת `AnimationDrawable` ו `View Animation`.
2. מטלת ביצוע לצור אנימציה ותגובה לאירוע המגע כהרחבה `SurfaceView`.

חלוקת שעות:

נושא	עיוני	התנסות
ציור בעזרת XML	1	2
ציור על <code>View</code> , <code>Canvas</code> , <code>Paint</code> , <code>BitmapFactory</code> .	2	4

4	2	ציור על SurfaceView, SurfaceHolder.
2	1	אנימציה ע"י AnimationDrawable, אנימציה בשיטת View Animation.
4	2	אנימציה ב SurfaceHolder, SurfaceView.

פרק 20 – הנחייה והגשת עבודת הגמר

הקדמה

בשלב זה סיימנו את החלק הלימודי ולתלמידים יש ידע נרחב כולל התנסות מעשית ודוגמאות ליכולות השונות של המערכת זה הזמן לקבל החלטות על הנושא ולממש את הרעיון בהתאם לשלבים שגלמדו - להתחיל בניתוח פרויקט ע"י תרשימים (תרשים מסכים ומחלקות ע"י UML) ולעבור לביצוע מעשי של הפרויקט.

מטרות כלליות:

ביצוע פרויקט הגמר, בהתאם לדרישות שנקבעו על פי הרמה המתוכננת.

מטרות ביצועיות:

1. התלמיד יסביר את ההבדל בין הרמות של עבודות הגמר.
2. התלמיד יסביר את דרישות וצרכי הרמה של עבודת הגמר שבחר לבצע.
3. התלמיד יבצע (ביחד עם המורה) ניתוח לפרויקט עבודת הגמר לרמות המפורטות בתכנים.
4. התלמיד יקבע את לוח הזמנים וסדר ביצוע הרכיבים של הפרויקט.
5. התלמיד יבצע את הפרויקט בהנחייה רציפה של המורה.
6. התלמיד יציג תיעוד לכל המחלקות והשיטות והתכונות.
7. התלמיד יצור תיק פרויקט לעבודת הגמר.

מושגים והכוונה:

1. הסבר לגבי ההבדלים בין הרמות של עבודות הגמר ודרישות החובה לכל רמה.
2. דגשים לבחירת נושא לפרויקט מלווה.
3. ביצוע ניתוח לפרויקט שכולל את הרמות הבאות:
 - i. רמת הממשק: הגרפיקה, מסכים ותרשים זרימה בין המסכים.
 - ii. רמת התכנות: זיהוי ישויות, עצמים ורכיבים ייחודיים של אנדרואיד. קביעת המחלקות הנדרשות (כולל המחלקות של ממשק המשתמש הגרפי), זיהוי אלה שקיימות וניתן להשתמש בהן ואלה שצריך ליצור.
- יצירת תרשים UML לתיאור המחלקות והקשרים ביניהם.
- iii. רמת הלוגיקה: זרימת פעילות היישום, חוקים וכדומה בפרט כשמדובר במשחקים.
- iv. רמת הנתונים: תיאור הנתונים שצריכים להישמר על המכשיר, דרך האחסון שנבחרה (הגדרות, קבצים, SQLITE).

- v. רמת השירותים והתכנות המקבילי: תיאור את הצורך ברכיבים או קטעי בקוד שצריכים לרוץ ברקע (שירותים לדוגמה) והצורך בתכנות מקבילי.
4. קביעת לוח זמנים שמתאר את סדר ביצוע הפרויקט החל מההצעה עד תיק הפרויקט.
5. ביצוע הפרויקט שנבחר עד תומו.
6. כתיבת תיעוד של הקוד. (רצוי לכתוב את התיעוד במהלך כתיבת הקוד)
7. כתיבת תיק הפרויקט.

דרכי הוראה:

על המורה להיות מעורב בתהליך כבר בשלב הצעת הפרויקט ולוודא שניתן לבצע אותו. חשוב להיות מעורב בעבודת התלמידים משלב הניתוח עד הביצוע.

באחריות התלמיד אבל גם המורה לוודא כבר בשלב ההתחלתי שהפרויקט יכול לעמוד בדרישות המינימאליות של רמת עבודת הגמר.

על המורה להכין הצעת פרויקט לדוגמה לשימוש התלמידים לאופן הגשת ההצעה.

המורה יבצע ניתוח לפרויקט ביחד עם התלמידים תוך כדי רישום וציור. המטרה לתוצר של מסמך שכולל תרשים זרימת מסכים, תרשים UML, רכיבים ייחודיים של אנדרואיד ותיאור הנתונים הנשמרים.

לצור תרשים שמקשר את כל הרכיבים והתרשימים שבוצעו בשלב שלפני ביחד.

להתמנות את המשך העבודה באישור המורה.

אפשר לבצע את שלב הניתוח של הצעות התלמידים בקבוצות, כך שכל קבוצה תבצע ניתוח של כל חברי הקבוצה.

בביצוע של הפרויקט מומלץ לאמץ את "מודל הקליפות" או מודל ספיראלי על פי השלבים:

- שלב 1: בכל שלב מגדירים יעד שניתן להשיג בזמן קצר באופן סביר ומתחילים ביישומ.
- שלב 2: כשמסיימים בודקים שהיעד הושג על ידי ביצוע סדרת בדיקות.
- שלב 3: יוצרים גרסה חדשה (עוזבים את הישנה ומתחילים משלב 1).

בשיטה זו הפרויקט יגדל שלב אחרי שלב "קליפה אחרי קליפה" או ספיראלית, ותמיד תהיה גרסה אחרונה יציבה ותקינה כך שאם מאבדים או מקלקלים את הגרסה האחרונה תמיד יהיה פרויקט בגרסה של שלב אחד לפני.

מומלץ להחזיק מחברת או קובץ ביצוע פרויקט, שכולל/ת את הניתוח והתרשימים, מעקב מתועד מסוג זה עוזר בתכנון הזמנים ובתכנון השלב הבא לביצוע ומאפשר ראייה ברורה מה עדיין נשאר לבצע. בתיעוד מציינים בכל פעם שניגשים להמשך ביצוע הפרויקט את מספר הגרסה והיעד, ובסוף מציינים אם הושג כן או לא, מסיקים מסקנות ואופטיביות ומתכננים ומציינים את הצעד הבא בגרסה הבאה.

מן הסתם, מומלץ שלמורה ידע א כל התכנים ותהיה לו דוגמה לכל תוכן לפני תחילת ביצוע הפרויקט עם התלמידים. אבל יש להבין שרשת האינטרנט עמוסה בתכנים כך שלעיתים קרובות המורה לא יכול להכיר את כולם. לכן תפקידו כמנחה ללמידה הוא קריטי!

דרכי הערכה אפשריות לפרק זה:

1. הערכת הדרך אותה עושה התלמיד - להעריך את עבודת התלמידים לפי השלבים שצריך להגיש: הצעת פרויקט, ניתוח וכו'. לתת לתלמידים משימות בשלבים.
2. לתת לכל תלמיד להעריך בעצמו את ביצוע כל שלב מהשלבים בפרויקט.
3. להעריך את מידת העמידה בלוח הזמנים אותו קבע התלמיד בעצמו למהלך הפרויקט.

חלוקת שעות:

נושא	עיוני	התנסות
הסבר מפורש על ההבדל בין הרמות	1	0
מחקר וניתוח פרויקט מלווה.	10	18
קביעת לוח זמנים	2	
ביצוע הפרויקט.	8	25
תיק פרויקט	3	5