ממשק Game: מטרתו להגדיר אילו שיטות יחויבו לממש המחלקות שמיישמות את הממשק, במקרה זה השיטות הן התחלת וסיום משחק. בעצם קובע את התנהגות המשחק הכללית מאוד.

מחלקת Snake: מחלקה שמכילה את המחלקה SnakeBodyPart, באוסף מסוג ArrayList, מטרת המחלקה היא לייצג את גוף הנחש, להיות אחראית על הלוגיקה לתזוזת הנחש על לוח המשחק בהתאם לקלט מהמשתמש, להגדיל (להוסיף חלק לגוף) הנחש במקרה שבו הנחש אוכל פרי (מתנגש באובייקט מסוג Fruit) ולבדוק התנגשות ראש הנחש בגוף הנחש (התנגשות עצית אשר תוביל להפסד במשחק).

הבנאי של המחלקה מאתחל את גוף הנחש (snakeBody) להיות ArrayList שתכיל אובייקטים מסוג SnakeBodyPart. ומוסיף חלק ראשון לsnakeBody בהתאם לפרמטרים X וY שהבנאי מקבל.

מחלקת Point: מחלקה אבסטרקטית, שלא ניתן לייצר מופע שלה, משמשת כblueprint למחלקות שמייצגות אובייקט על לוח המחשק. מכילה תכונות לקואורדינטות x וy, בנאי אשר מקבל X ו-Y ומאתחל את התכונות בהתאם, getters וsetters לתכונות.

המחלקות המרחיבות את Point יורשות את המתודות והשיטות שלה ויכולות לדרוס/ להעמיס מתודה מסוימת או להוסיף התנהגות/תכונה בהתאמה אישית ליצירת אובייקטים ללוח המשחק, למשל במחלקת SnakeBodyPart שיורשת מPoint נוספה תכונה פרטית שמייצגת את התו של חלק גוף הנחש.

מחלקת SnakeBodyPart: מייצגת חלק בגוף הנחש אשר יוצג על לוח המשחק, יורשת ממחלקת Point.

מחלקת Fruit: מייצגת את הפרי אשר יוצג על לוח המחשק, יורשת ממחלקת Point.

מחלקת GameBoard: מחלקת המייצגת את לוח המשחק, מכילה תכונות המייצגות את ממדי הלוח (במקרה זה הלוח הוא בגובה 20 שורות וברוחב 20 עמודות), הלוח מיוצג על ידי מערך דו ממדי של תווים. הבנאי שלה מאתחל מערך דו ממדי בהתאם לממדי הלוח (התכונות) וממלא את הלוח בתווים ריקים על מנת לשמור על תצוגה אחידה של הלוח בכל פלטפורמה (יתכנו תכונות מסוימות אשר יציגו לוח לא נקי למרות שבהגדרת מערך דו ממדי של תווים ברירת המחדל לתו שלא אותחל מערך הוא תו ריק).

המחלקה מכילה מתודות להדפסת לוח המחשק, למיקום אובייקט הפרי ואובייקט הנחש על לוח המחשק בהתאם לקואורדינטות שלהם, כאשר במיקום השורות נשתמש בקואורדינטה Y ובמיקום העמודות נשתמש בקואורדינטה X.

מחלקת SnakeGame: מייצגת את המשחק עצמו. מיישמת את הממשק Game ומיישמת את השיטות stop ו-start של הממשק: בתחילת המשחק יוצגו הנחיות לשחקן, ימוקמו הפרי והנחש על הלוח והלוח יודפס לקונסולה. בסיום המשחק יודפס "Game Over" והמשחק יופסק.

מכילה תכונות מסוג Snake לייצוג מופע של מחלקת הנחש, SnakeBodyPart לייצוג מופע של ראש הנחש (לצורך בדיקת התנגשות של הנחש עם הפרי), Fruit המייצגת מופע של מחלקת הפרי, משתנה עזר בוליאני gameOver וGameBoard המייצג מופע של לוח המשחק.

הבנאי של המחלקה מאתחל את snake (מייצר מופע של Snake) ומספק את מיקום לחלק הראשון של הנחש (10*10), מגדיר את ראש הנחש head כאובייקט באינדקס ה0 ברשימת שמייצגת את הנחש. מאתחל את fruit (יוצר מופע של Fruit) ומספק את המיקום ההתחלתי לפרי בלוח המשחק (5*5), מאתחל את משתנה העזר gameOver להיות FALSE ותאתחל את לוח המחשק כמופע של GameBoard.

מכילה שיטות GET וSET לתכונות המחלקה ואת השיטות הבאות:

- changeFruitPosition אחראית לשינוי מיקום הפרי לאחר שהנחש אוכל את הפרי, תוך ביצוע בדיקות למציאת מיקום חדש - כך שהמיקום לא יקבל משבצת שבה יש חלק גוף מסוים של הנחש, המיקום נקבע רנדומלית בעזרת מימוש מחלקת Random. לאחר מכן היא מגדירה את הקואורדינטות של הפרי בעזרת SETTERS וממקמת את הפרי במיקומו החדש על לוח המשחק.

- checkFruitCollision בודקת האם הייתה התנגשות בין ראש הנחש לפרי על לוח המשחק (השוואת קואורדינטות YוX של הנחש ושל הפרי (האם נמצאים באותו תא במערך הדו ממדי)), במידה וכן מחזירה TRUE וFALSE אחרת.
- checkSnakeCollision בודקת האם הייתה התנגשות של הנחש בעצמו תוך שימוש בשיטה checkSelfCollision במחלקת Snake שמחזירה משתנה בוליאני.
- updateGameState אחראית על עדכון המשחק בהתאם לקלט מהמשתמש, הזזת הנחש למיקום החדש, בדיקת התנגשות עצמית של הנחש בעצמו ובדיקת סטטוס המשחק (gameOver) במידה והייתה התנגשות של הנחש בעצמו, בדיקת התנגשות הנחש בפרי (אכילת פרי) – במידה וכן הפרי ישנה מיקום, הנחש יגדל (יתווסף חלק לגוף הנחש), הנחש והפרי ימוקמו מחדש בלוח, הלוח יודפס לקונסולה ונצא מהאיטרציה הנוכחית (מחלקת MAIN רצה לולאה בהתאם לסטטוס המשחק) ונחכה לקלט הבא מהמשתמש לקבלת כיוון הזזת הנחש.במידה ולא הייתה התנגשות בפרי, ננקה את הלוח (כדי לוודא מחיקת מיקום אחרון של זנב הנחש – כיוון הנחש התקדם) נמקם את הנחש מחדש על הלוח, נמקם את הפרי על הלוח ונדפיס את הלוח.

מחלקת MAIN: מייצרים מופע של scanner לקבלת קלט מהמשתמש, מייצרים מופע של משחק SnakeGame, מריצים את המשחק ()game.start, ונכנסים ללולאה שהתנאי עצירה שלה הוא כאשר משתנה העזר gameOver של מחלקת SnakeGame יהיה TRUE,

בכל איטרציה ננקה את הקונסולה, נקבל קלט מהמשתמש שישמר במשתנה userInput מסוג char, ונריץ את המתודה updateGameState של מחלקת SnakeGame המקבלת כפרמטר את הקלט של המשתמש. בסוף כל איטרציה נדפיס למשתמש הוראה להזנת תו בהתאם לכיוון שבו ירצה להזיז את הנחש.

במקרה שהמשתנה gameOver יהפוך במהלך כלשהו לTRUE תופעל המתדה stop אשר במחלקת SnakeGame והמשחק יפסק.