**Group Number 1 / Date: 25.4.16**

|  |  |  |
| --- | --- | --- |
| shayscal90@gmail.com | 302632105 | Shay Shahar |
| yakir1114@gmail.com | 312143969 | Yakir Karandian |
| rizraz2001@gmail.com | 305003659 | Raz Cohen |
| [asaff19899@gmail.com](mailto:asaff19899@gmail.com) | 200940997 | Assaf Tzarfati |

**Assignment 2 Answers:**

1-

2-

א.

אחד העקרונות של Object Oriented Analysis & Design שבו השתמשנו הוא

הורשה: עקרון ההורשה מהווה שימוש בתכונות של מחלקה אחת במחלקה אחרת תוך הוספה של תכונות ו/או מתודות אחרות, הייחודיות למחלקות אחרות.

עיקרון זה בא לידי ביטוי בכמה מקומות ב-Class Diagram:

מחלקת GeneralUI שמורישה לשאר מחלקות ה-GUI הנמצאות בתרשים המחלקות. הורשה זו לכלל מחלקות הבנים לטובת הממשק הגרפי.

בנוסף מחלקת AbstractReport מורישה למחלקת WeeklyReport למשל ו-Monthly Report יורשת מ-WeeklyReport כיוון שאלו תכונות שצריכות להופיע גם במחלקה היורשת. בנוסף יש את המחלקה Person שכל שחקן במערכת שהוא אדם יורש ממנה. מחלקה זו מייצגת את תכונות האדם הבסיסיות שיש לשמור לכל אדם במערכת.

בנוסף, קיימת הורשה ממחלקת- AbstractClient וממחלקת- AbstractServer

למחלקות ClientController, ServerController בהתאמה. הורשה הזו תבוא לידי ביטוי בשלב המימוש ברגע בו נצטרך ליצור את הקשר בין ה-Client ל-Server.

ב.

פולימורפיזם זו תכונה של [שפות תכנות](http://he.wikipedia.org/wiki/%D7%A9%D7%A4%D7%AA_%D7%AA%D7%9B%D7%A0%D7%95%D7%AA) מונחות עצמים המאפשרת לטפל ב[ערכים](http://he.wikipedia.org/wiki/%D7%A2%D7%A8%D7%9A_(%D7%9E%D7%93%D7%A2%D7%99_%D7%94%D7%9E%D7%97%D7%A9%D7%91)) מ[טיפוסים](http://he.wikipedia.org/wiki/%D7%98%D7%99%D7%A4%D7%95%D7%A1_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) שונים בעזרת [ממשק](http://he.wikipedia.org/w/index.php?title=%D7%9E%D7%9E%D7%A9%D7%A7_%D7%AA%D7%95%D7%9B%D7%A0%D7%94&action=edit&redlink=1) אחיד. שפת תכנות התומכת בפולימורפיזם נותנת ל[מתכנת](http://he.wikipedia.org/wiki/%D7%9E%D7%AA%D7%9B%D7%A0%D7%AA) את היכולת לממש [אלגוריתמים](http://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D) ו[מבני נתונים](http://he.wikipedia.org/wiki/%D7%9E%D7%91%D7%A0%D7%94_%D7%A0%D7%AA%D7%95%D7%A0%D7%99%D7%9D) לשימוש כללי, ולגזור מהם צורות שימוש שונות בהתאם לעצמים ולנסיבות המשתנות.

במערכת שהתבקשנו לממש אין אובייקטים מטיפוסים שונים שיש לטפל בהם ע"י ממשק אחיד לכן עקרון זה אינו בא לידי ביטוי.