**חלק יבש**

סעיף 1)

Event

Mtmchkin

Player

Responsible

Jobs

Personality

Risk  
Taking

Sorcerer

Warrior

Encounter

Goblin

Dragon

Card

Potion  
Merchant

Solar Eclipse

Gang

Giant

cards

\*

players

\*

job

personality

members

\*

סעיף 2)

השתמשנו בDesign Pattern משני סוגים, ניתן גם לראות אותם בדיאגרמת הUML מסעיף הקודם. הסוג הראשון הוא Strategy, השתמשנו בו במחלקת Player עבור הpersonality וגם עבור Jobs.   
הסוג השני הוא composite, שמשומש במחלקת הMtmchkin וגם במחלקת הGang.

סעיף 3)

ניתן לעשות זאת בכמה דרכים, נבחר ונסביר את השיטה הבאה:  
תחילה נוסיף Job חדש בשם הNinja שיורש מJobs ואין צורך לשנות את הפונקציות שכבר קיימות. עם זאת נוסיף מיתודה חדשה escape לJobs שמחזירה true אם הplayer מתחמק מהקרב כך שבJob הדיפולטי הפונקציה תמיד תחזיר שלא (false) אך עבור הninja נעשה override כך שהיא תבדוק אם הוא צריך להתחמק או לא לפי התנאים הנתונים.   
עוד שינוי דרוש הוא במחלקה Encounter בפונקציה playCard() נוסיף if בהתחלה שתבדוק אם הplayer יתחמק או לא, אם כן אז נסיים את הרצת הפונקציה ללא השפעה על הplayer.  
וגם נשנה בהתאם את הgetJob() עבור הninja שיחזיר “Ninja”.

סעיף 4)

ניתן לממש את זה בקלות במערכת שלנו מכיוון שהjob הוא שדה אצל הplayer שממומש ע"י strategy pattern לכן ניתן לשנותו בקלות. כדי לעשות את זה:  
נוסיף מיתודה עבור הplayer שתבחר job רנדומלי ואז בעזרת הפונקציה setJob() שקיימת אצלנו נשנה את הJob. נוסיף גם את הDivineInspiration שיורש מהEvent כך שהplayCard() שלו קורא לפונקציה שנזכרה לפני.  
וגם עבור ההדפסות יש צורך לשנות את הgetDescription() בהתאם ולהוסיף את ההדפסות של הEvent לתוך הplayCard().