

דיאגרמת Sequences

הקדמה לדיאגרמת Sequences

הודעות בין אובייקטים

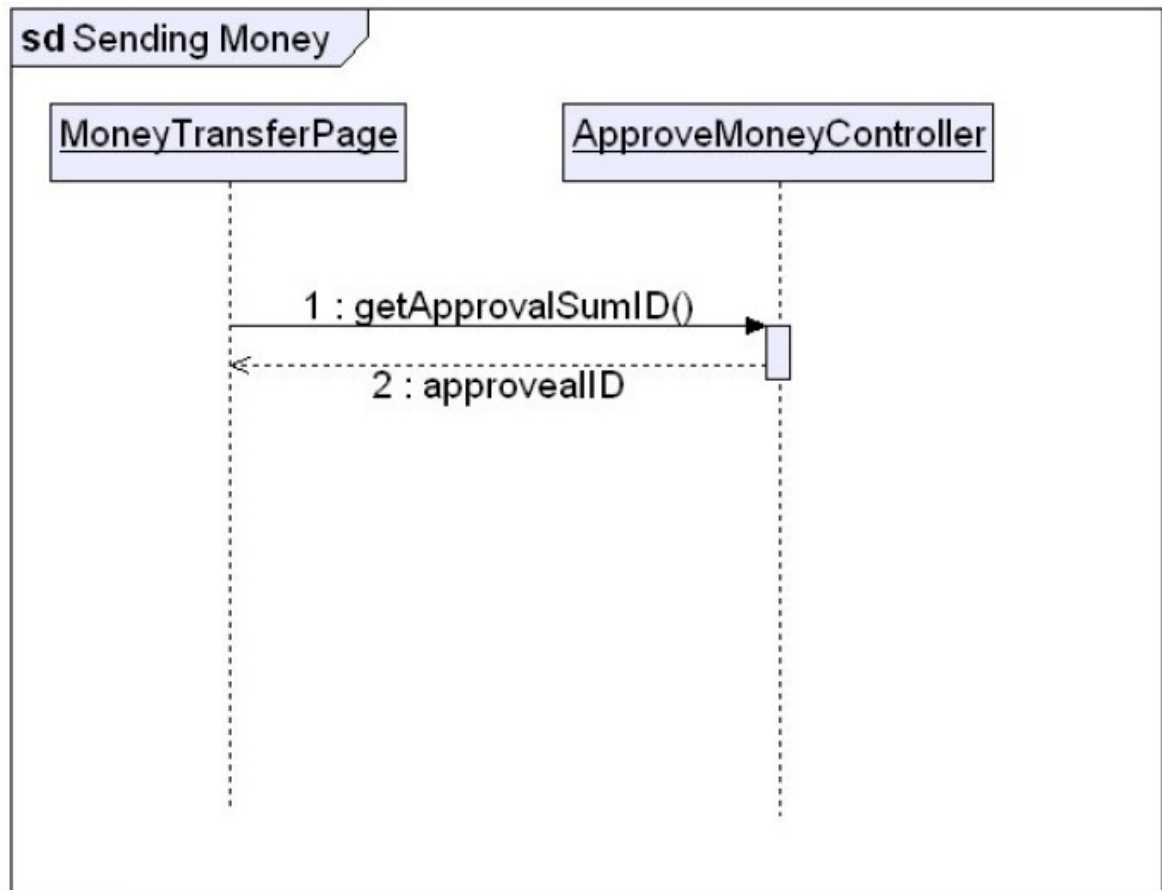
ה – Execution Occurrences

שימוש ב – State Invariants

הקדמה לדיאגרמת Sequences

ה – Sequence Diagrams שייכות לקבוצה גדולה יותר של דיאגרמות בשם Interaction Diagrams. באמצעות Sequence Diagram ניתן לתאר את התקשורת בין האובייקטים שיש במערכת שמפתחים. דיאגרמות מסוג Sequence Diagram מתמקדות ב-messages השונים שולחים אחד לשני.

בדיאגרמה הבאה מתוארים האובייקטים MoneyTransferPage ו- ApproveMoneyController. בין שני האובייקטים הללו מתנהלת תיקשורת באופן הבא: אובייקט מסוג MoneyTransferPage שולח לאובייקט מסוג ApproveMoneyController הודעה שבה הוא מבקש אישור לבצע העברה בסכום מסוים. אובייקט ה- ApproveMoneyController שולח בחזרה הודעה אשר כוללת את מספר האישור שניתן.



כל אובייקט מתואר באמצעות מלבן שבתוכו כותבים את שמו של האובייקט בפורמט הבא:

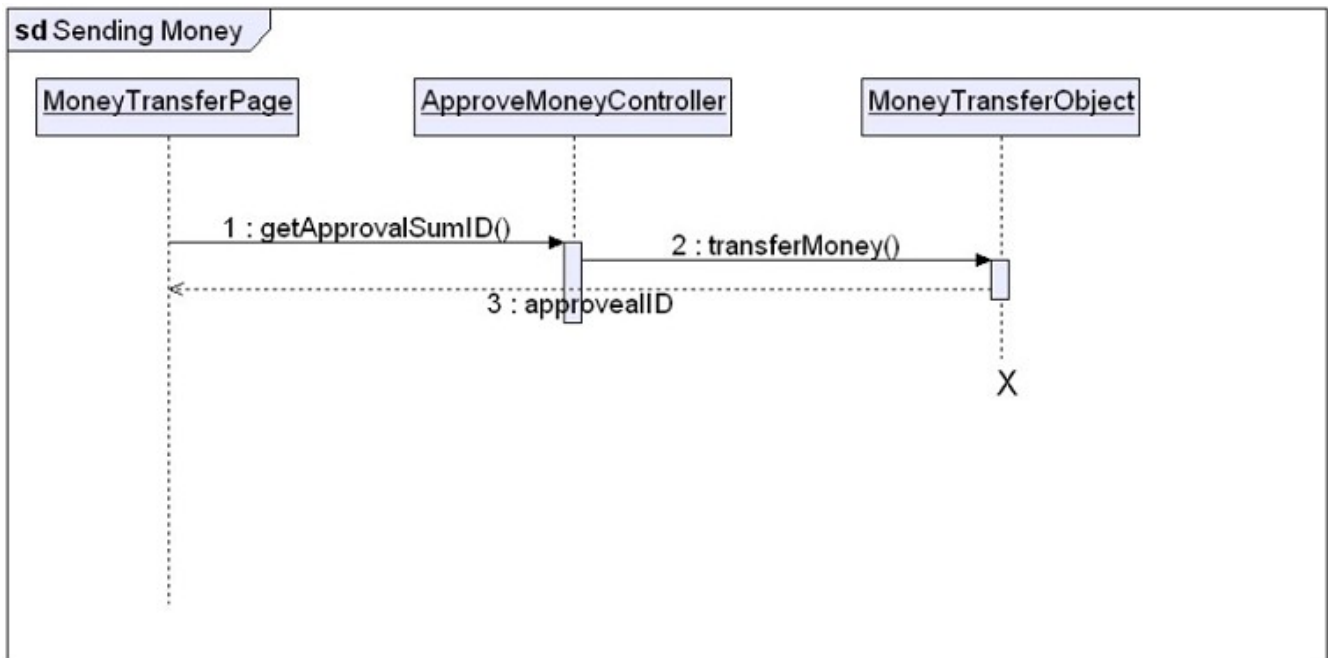
`object_name [selector] : class_name ref another_diagram`

ה – selector הוא אופציונלי והוא משמש באותם מקרים שבהם יש לנו מערך של אובייקטים (או מבנה נתונים אחר שמחזיק סידרה של אובייקטים). ציון שם ה-class שממנו נוצר האובייקט גם הוא נחשב לאופציונלי. כאשר קיים תיאור מפורט יותר (בדיאגרמה אחרת) של האופן שבו אובייקט מסויים פועל ניתן להפנות לתיאור המפורט הנ"ל באמצעות כתיבת שמה של הדיאגרמה האחרת בצירוף המילה 'ref'. מתחת לשם האובייקט יש למתוח קו תחתי.

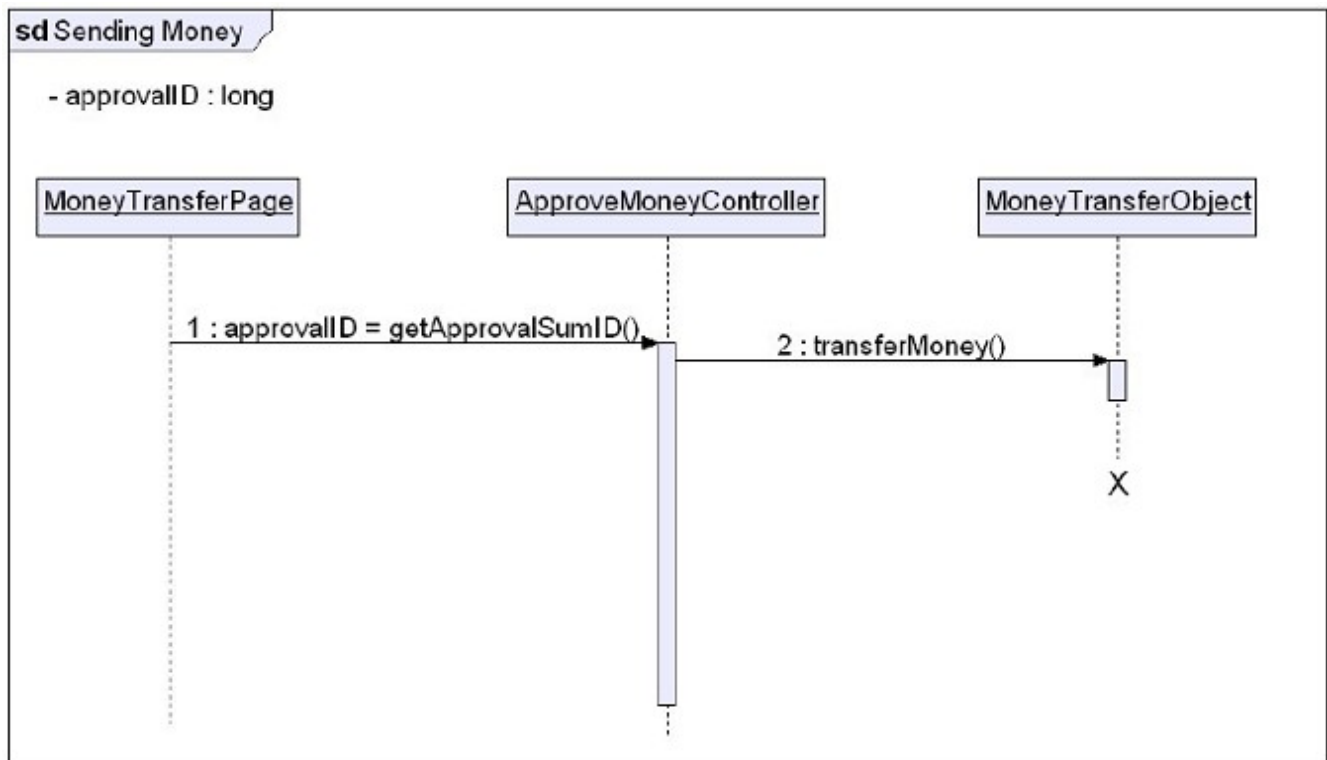
ה – Life Line הוא קו מקווקו אשר יורד מטה מכל מלבן שמתאר אובייקט בדיאגרמה. ה- Life Line מבטא את משך חייו של האובייקט.

את ה- Sequence Diagram ממקמים בתוך frame שבחלקו העליון השמאלי כותבים את שם הדיאגרמה בצירוף האותיות "sd" (ראשי תיבות של: Sequence Diagram).

כדי לתת ביטוי לסוף חייו של אובייקט מסויים מוסיפים את הסימן X על גבי ה- Life Line שלו. הדוגמה הבאה מציגה זאת.



ניתן להוסיף לדיאגרמה משתנים מקומיים. יש לפרט אותם בחלק השמאלי העליון של הדיאגרמה באופן הדומה לתיאור משתנים שמוגדרים ב-Class Diagram.



ה – messages שנשלחים בין אובייקטים

התקשורת בין האובייקטים שנוטלים חלק בדיאגרמה מתבצעת באמצעות messages שנשלחים ביניהם. ה-messages יכולים להיות כל אחד מהסוגים הבאים:

קריאה להפעלת מתודה על אובייקט אחר

יצירה של אובייקט חדש (האובייקט האחר)

הריסת האובייקט (האובייקט האחר)

כאשר ה-message הוא למעשה קריאה להפעלת מתודה על האובייקט האחר וקריאה זו כוללת את שליחתם של ארגומנטים ניתן לתת לכך ביטוי באמצעות כתיבתם בתוך גבולות הסוגריים העגולות שמופיעות ב-message.

הפורמט שבו יש לכתוב message שמהווה קריאה להפעלת מתודה הוא:

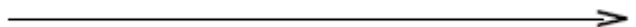
`attribute = operation_name (arguments) : return_value`

ה-attribute הוא המשתנה שלתוכו ייכנס הערך שמוחזר על ידי המתודה המופעלת. באמצעות ציון ה-return_value ניתן לתאר את הטיפוס של הערך המוחזר על ידי המתודה המופעלת. כל message אשר נשלח על ידי אובייקט אחד לאחר יכול להיות synchronous או asynchronous.

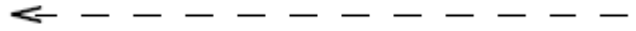
כאשר ה-message הוא synchronous אז המשמעות היא שהאובייקט שמבצע אותו ימתין עד לסיומו ורק אז ימשיך (כלומר, אם אובייקט מסויים קורא להפעלת מתודה על אובייקט אחר באופן שהוא synchronous אז הוא ימתין עד שהמתודה תסתיים ותחזיר את הערך לפני שהריצה של התכנית תימשך). כדי לתאר synchronous message יש להשתמש בקו חץ רציף עם ראש חץ סגור.



כאשר ה-message הוא asynchronous אז המשמעות היא שהאובייקט שמבצע אותו לא ימתין לסיומו וימשיך באופן מיידי (כלומר, אם אובייקט מסויים קורא להפעלת מתודה על אובייקט אחר באופן asynchronous אז הוא כלל לא ימתין לקבלת הערך המוחזר מהמתודה וימשיך באופן מיידי בתכנית). כדי לתאר asynchronous message יש להשתמש בקו חץ רציף עם ראש חץ פתוח.



את הערך המוחזר ניתן לתאר באמצעות קו חץ מקווקו עם ראש חץ פתוח באופן הבא:



כדי לתאר message שמהווה יצירה של אובייקט יש להשתמש בקו מקווקו עם חץ פתוח אשר מצביע על האובייקט שנוצר ולהוסיף מעליו את הטקסט הבא:

create (class name)

כך למשל, התרשים הבא מתאר message שמשמעותו יצירה של אובייקט מטיפוס SportCar.

create (SportCar)



כדי לתאר message שנשלח ומעולם לא הגיע ליעדו יש להשתמש בעיגול מלא אשר אליו החץ יצביע.



כדי לתאר message שהגיע מבלי שיהיה ברור מי שלח אותו יש להשתמש בעיגול מלא אשר ממנו החץ יצא.



ה – Execution Occurrences

ה-Execution Occurrence מתאר אובייקט אשר מעורב בביצוע פעולה/פעולות. ניתן לדמיין את ה-Execution Occurrence כמי שמייצג את מעורבותו של האובייקט בפעילותו של thread מסוים.

את ה-Execution Occurrence מייצגים באמצעות מלבן אפור שמציירים מעל הקו המקווקו.



שימוש ב – State Invariants

ה- State Invariant הוא תנאי לוגי (בדרך כלל ביטוי מטיפוס boolean) אשר כותבים בתוך סוגריים מסולסלות מעל Life Line נתון, והוא מתאר מצב טבע שחייב להתקיים לכך אורך המשך ה- interaction. כיוון ביצועו של interaction הוא משמאל לימין מלמעלה למטה. בכל פעם ש- message מגיע ל- Life Line האמור התנאי נבדק מחדש ובמידה שהוא false ה- execution האמור נפסק.

