

הסבר תיקון

לא התקבלה התוצאה הרצויה שינוי ערך ה-x של p2 שינה בנוסף את ערך ה-x של p1, בנוסף בחלק מן ההרצות התוכנה קורסת.

ראשית על מנת להסביר את התיקון, נצטרך להסביר את הבעיה. בשורה השנייה בפונקציית ה-main מצהירים על אובייקט חדש מסוג נקודה בשם p2 ובתור ערכו משימים את p1 שגם הוא אובייקט מסוג נקודה.

בשפה ++c כאשר מצהירים על אובייקט חדש ובאותה שורה מתבצעת השמה לערכו, הבנאי (constructor) ש"נקרא לבניית האובייקט" הוא הבנאי ההעתקה.

בקוד הנ"ל יש מימוש לבנאי ההעתקה, כמו שמצוין בעמוד 28 במדריך הלמידה יש 2 סוגים של העתקות אובייקטים, העתקה עמוקה והעתקה רדודה. הבנאי שלנו מעתיק את המצביע של התכונה _coord ולא את הערכים שבתוכו משמע – העתקה רדודה. כאשר אנו מתעסקים באובייקטים מורכבים, כלומר אובייקטים שכוללים מצביעים לרכיבים כלשהם המהווים חלק מהאובייקט (במקרה שלנו המערך הדינאמי _coord) יש צורך בהעתקה עמוקה. בפועל המימוש של בנאי ההעתקה בקוד שלנו הוא העתקה רדודה וכתוצאה מההעתקה הרדודה מתקיימים 2 באגים.

1. בשורה p2.setX(5) אשר מטרתה לשנות את ערך ה-x של p2, חוץ מערך ה-x של p2 אשר משתנה ל-5, גם ערך ה-x של p1 משתנה ל-5 כיוון שהם חולקים אותו מקום בזיכרון בעקבות ההעתקה הרדודה.

2. אנו מנסים לעשות delete למצביע כבר שנעשה לו delete (בעקבות ההצבעה הכפולה מתבצע delete כפול) התנהגות זו אינה מוגדרת ועלולה (וכנראה) להקריס את התוכנה.

ולכן על מנת לתקן שגיאות אלו, נשנה את מימוש בנאי ההעתקה כך שיבצע ההעתקה עמוקה באמצעות יצירת מערך דינאמי חדש עבור האובייקט שאנו מעוניינים להעתיק את המידע אליו והשמת הנתונים של האובייקט המועתק לתוך המערך החדש של האובייקט החדש.

הערה

ראוי לציין כי כאשר אנו קוראים לdestructor של point אנחנו משתמשים בפקודה delete כדי למחוק את הקצאת המערך של התכונה _coord כשבפועל אנו אמורים להשתמש בפקודה delete[]. כתיבת קוד שכזאת אינה נכונה ועלולה לגרום להתנהגות לא צפויה, אם זאת במקרה שלנו התנהגות זאת אינה פוגעת בתוכנית והכל מתבצע כנדרש.