

מטלה 9

במטלות הקודמות שמתי לב שרבים מכם לא אוהבים לכתוב בדיקות אוטומטיות. זה חבל מאד - אתם מפספסים את התענוג של לתת למחשב לגלות עבורכם באגים...

כדי לראות כמה זה קל לכתוב בדיקות אוטומטיות, במטלה הנוכחית נפתח מערכת לבדיקות אוטומטיות כלליות. עליכם לכתוב מחלקה בשם `TestCase` עם השיטות הבאות:

- `check_equal` - בודקת אם שני הארגומנטים שלה שווים, אם לא - מדפיסה הודעת שגיאה.
- `check_different` - בודקת אם שני הארגומנטים שונים, כנ"ל.
- `check_output` - בודקת את אופרטור הפלט של הארגומנט שלה, משווה את התוצאה למחרוזת נתונה.
- `check_function` - בודקת פונקציה כללית כלשהי עם ארגומנט אחד.
- `print` - מדפיסה את תוצאות הבדיקה - כמה בדיקות עברו ונכשלו.
- **בנאי** - הבנאי מקבל שני פרמטרים. הפרמטר ראשון הוא שם הבדיקה (מחרוזת) - שם הבדיקה מודפס לפני כל הודעה היוצאת מהמחלקה. הפרמטר השני הוא זרם-פלט (`ostream`) - כל ההודעות היוצאות מהמחלקה נכתבות לשם.
- השיטות צריכות לעבוד עם משתנים **מכל** סוג - גם סוגים פרימיטיביים כמו `int` וגם מחלקות חדשות שמגדירים. ראו דוגמה בתוכנית `main` המצורפת.
- יש לוודא שבגיטהאב שלכם יש שני קבצים - `TestCase.cpp`, `TestCase.hpp` (אפשר להוסיף קבצים נוספים אבל הם לא ייכללו בקימפול האוטומטי).
- במטלה זו אין להשתמש בספריות חיצוניות לבדיקות אוטומטיות - המטלה היא לכתוב את הספריה בעצמכם.

דגשים:

- יש לחזור על החומר של ההרצאות לפני שמתחילים לכתוב, ולהשתמש בו לפי הצורך.
- יש להקפיד על כללי הנדסת תוכנה, ובפרט: קוד קריא, תיעוד ובדיקות.
- מותר להשתמש בתכונות מתקדמות של שפת C++ גם אם עדיין לא נלמדו בהרצאות.
- מותר להשתמש בקוד מהאינטרנט, אולם: (א) יש לציין בבירור את המקור, (ב) יש לבדוק שהקוד שהורדתם עובד בלי שגיאות, (ג) יש להבין את הקוד שהורדתם כך שתוכלו להסביר אותו בכיתה.