מטלה – ירושה, בול-פגיעה

במטלה זו נלמד על דגם-עיצוב שנקרא "אסטרטגיה", שהוא אחד השימושים העיקריים למנגנון הירושה.

המשימה שלכם היא לכתוב סוכנים אוטומטיים למשחק בול-פגיעה.

עליכם לממש את:

- הפונקציה calculateBullAndPgia – פונקציה המחשבת את מספר הבולים והפגיעות בניחוש נתון (שימו אותה בקבצים calculate.hpp/calculate.cpp).
- המחלקה Chooser – מחלקה מופשטת המייצגת את השחקן הבוחר מספר.
- המחלקה Guesser – מחלקה מופשטת המייצגת את השחקן הצריך לנחש את המספר.
- המחלקה SmartGuesser – סוכן אוטומטי היודע לנחש כל מספר תוך 100 צעדים לכל היותר.

קבצים

מצורפים למטלה זו הקבצים הבאים:

- הקבצים play.hpp/cpp – מגדירים את הפונקציה play המריצה משחק בול-פגיעה יחיד.
- הקבצים DummyChoosers.hpp/cpp – מגדירים כמה סוכנים אוטומטיים לבחירת מספר (לא חכמים במיוחד).
- הקבצים DummyGuessers.hpp/cpp – מגדירים כמה סוכנים אוטומטיים לניחוש מספר (גם לא חכמים במיוחד).
- Demo.cpp – תוכנית ראשית לדוגמה.
- Test.cpp – תוכנית ראשית הכוללת בדיקות לדוגמה.
- הקבצים CompileError1/2.cpp – שתי תוכניות ראשיות שאמורות **לא להתקמפל** (ליצור שגיאת-קומפילציה).
- Makefile – קובץ ליצירת תוכנית הדוגמה ותוכנית הבדיקה.

שלבי העבודה

בשלב ראשון, עליכם לכתוב את הקבצים הדרושים על-מנת שהפקודות הבאות ירוצו בלי שגיאות קימפול:

```
make demo && ./demo
make test && ./test
```

בשלב זה אין לשנות את הקבצים הנתונים – עליכם לוודא שהתוכנית שלכם עובדת עם הקבצים הנתונים כמו שהם. כמו כן, לא חייבים לכתוב תוכנית המקבלת 100 בכל הבדיקות – רק שתתקמפל בלי שגיאות.

לאחר מכן, יש להרחיב את הקובץ Test.cpp ולהוסיף בדיקות-יחידה נוספות באותו סגנון של הבדיקות הקיימות (לא למחוק את הקיימות). יש לכתוב בדיקות-יחידה מפורטות. שימו לב – בשלב זה הקוד שכתבתם כנראה לא יעבור את כל הבדיקות – זה בסדר. העיקר שהבדיקות שלכם יהיו מלאות.

יש להגיש (במודל ובבדקן האוטומטי) את הקוד במצב זה – קוד שמתקמפל, וקובץ Test.cpp הכולל בדיקות-יחידה מפורטות, שעדיין לא כולן עוברות.

בשלב שני, יש לממש את המחלקות כך שיעברו את כל הבדיקות – גם הבדיקות שלכם וגם הבדיקות האוטומטיות שלנו.

יש להגיש תוך שבוע נוסף (במודל ובבדקן האוטומטי) את הקוד המלא.

בשלב שלישי, ייתכן שנעשה תחרות בין הסוכנים האוטומטיים שלכם (ה- SmartGuesser) וניתן בונוס לסטודנטים שהסוכנים שלהם יצליחו לפצח קודים במספר-התורות הנמוך ביותר. אם אתם מעוניינים להשתתף בתחרות זו, חשבתם על אסטרטגיה גאונית ואתם לא רוצים שיעתיקו אותה – אל תשימו אותה בינתיים בגיטהאב; אנחנו עובדים על שיפור לבדקן, שיאפשר לכם להגיש מאגר-גיטהאב פרטי.

הגשה לבדיקה אוטומטית

צרו מאגר (repository) חדש בגיטהאב והעלו לשם את הקבצים בתיקיה הראשית.

הגישו בטופס-ההגשה קישור-שיבוט למאגר - הקישור שרואים כשלוחצים על הכפתור clone בגיטהאב.

דגשים

- יש להקפיד על כללי הנדסת תוכנה, ובפרט: קוד קריא, תיעוד ובדיקות-יחידה מקיפות.
- **יש לחזור על החומר של ההרצאות לפני שמתחילים לכתוב, ולהשתמש בו לפי הצורך.**
- מותר להשתמש בתכונות מתקדמות של שפת C++ גם אם עדיין לא נלמדו בהרצאות.
- אין להעתיק תרגילים שלמים מסטודנטים אחרים. מותר להיעזר בקטעי קוד מהאינטרנט, אולם **יש לציין בבירור את המקור**, לוודא שהקוד עובד, ולוודא שאתם מבינים למה הוא עובד.