

סדנא בתכנות מונחה עצמים ושפת ++C - דף הנחיות לפינוקי הבית בקורס

יש להתעדכן באופן רציף בהודעות במערכת ה- moodle לגבי הבהרות בדף המטלה וכן יתכן וחלקים 2+3 יתעדכנו בהמשך.

דרישות הקורס

בקורס יש בחינה ופרויקט (אין בוחן). את התרגילים יש להגיש בזוגות.

התרגיל הינו "מתגלגל" כך שהחלק השני מתבסס על החלק הראשון.

את התרגיל השני אתם תמשיכו על בסיס פתרון של סטודנט אחר בקורס. לכן הקפידו שהקוד שלכם יהיה ברור, מתועד, כתוב נכון וידידותי למתכנת שיקבל אותו אחריכם.

לא ניתן בשום מקרה להמשיך את הפרויקט שלכם, אז אנא תחסכו ממני את השאלה בהמשך. וגם לא ניתן להגיש בשלשות או ביחידים.

ציון הפרויקט הינו ציון אחד עבור כל חלקיו.

הנחיות להגשת התרגילים

כל תרגיל יוגש לתיקיית google-docs משותפת בתאריך המבוקש, ויוצג במרוכז למרצה כחודש לאחר תום הסמסטר.

קישור לתיקיית ההגשה:

https://drive.google.com/drive/folders/1K47wfZfp_q8yfDcXw1x2ssglbjbKk5_z?usp=sharing

הערה כללית: חלק מהעבודה בתרגילים היא קבלת החלטות בנושאים שאינם מפורטים במדויק. המטרה היא לתרגל את הנושאים המרכזיים הנלמדים בקורס, ולא לתפור מוצר לפי דרישות של לקוח, ולכן בכל מקום שלא מופיעה דרישה מדויקת – מוטל עליכם לבחור בדרך ההגיונית ביותר שנראית לכם. כדאי מאוד שהיא תראה גם הגיונית בעיניי (-);

הנחיות כלליות (לכל חלקי הפרויקט)

- יש להימנע בכל מקרה משכפול קוד, ממתודות ארוכות מדי ומבחירת שמות גרועים למחלקות לפונקציות ולמשתנים.
- יש לוודא קלט תקין מהמשתמש בכל שלב ולהחזיר הודעות שגיאה קריאות במידה והקלט אינו תקין.
- יש להקפיד על מוסכמות בסגנון הכתיבה – שמות מחלקות יתחילו באות גדולה, משתנים ופונקציות באות קטנה, שמות קבועים יהיו מורכבים רק מאותיות גדולות, יש להקפיד על אינדנטציה נכונה, וכו'.
- יש לדאוג שבאף מקרה לא ייזרק Exception שלא טופל ושייגרם לתעופה של התוכנית.
- יש להקפיד על קוד פשוט ונקי.
- ייתכן דגש על תכנות מכון עצמים, ולכן שימו לב מי האובייקט האחראי על פעולה מסויימת, כלומר, אם יהיה באג, לכותב של איזו מחלקה יפנו באופן אינטואיטיבי, וכן שימוש נכון במנגנון החריגות.
- ומניסיון העבר: זהו קורס בשפה מכוונת עצמים, לכן הקפידו להשתמש בכל מאפייניה כפי שלמדתם בעבר.

הגדרת הפרוייקט

חלק 1

- א. עליכם להגדיר עולם בעיה המכיל לפחות את רכיבי התכנות הבאים:
- 10 מחלקות
 - 3 ירושות, לפחות אחת מרובה
 - 2 מחלקות עם הקצאות דינאמיות
 - 2 מחלקות שמכילות מחלקות אחרות
 - 2 מחלקות עם מערכים
 - לפחות 5 אופרטורים בנוסף לאופרטור =
 - 8 פעולות ב- main. שימו לב שפעולות אלו יאפשרו שימוש נוח ובסיסי במחלקות שכתבתם, הכולל הכנסה והצגה של הנתונים.

יש לצרף בנוסף לטקסט המתאר את הסיפור גם class diagram, המציגה את המחלקות, הורשות והכלולות.

את דף הסיפור המפורט יש להגיש לתיקיית ה- google-docs המיועדת עד יום שני ה- 12.08.2019 בשעה 23:59. שם קובץ ההגשה יהיה שמות המגישים ושם הסיפור, למשל: GogoCohen_And_MomoLevi – College.

את הערותי והארותי לסיפורים אכתוב ישירות על הקבצים שתגישו או פרונטלית בזמן התרגול. עליכם להתעדכן באופן תדיר ולראות האם כתבתי הערות, עד אשר אכתוב לכם במסמך שהגרסה סופית ומאושרת. יש להעלות רק מסמכי google docs ולא מסמכי word חיצוניים.

- ב. עליכם להגדיר את כל קבצי ה- H של הפרוייקט. יש להקפיד על כתיבת c'tor נכונים, copy c'tor ו- d'tor במידה הצורך, שיטות set ו- get, שימוש נכון ומוקפד ב- const. אין להשתמש ב- STL או ב- templates בחלק זה.
- ג. ממשו את ה- main עבור הפעולות שהגדרתם. שימו לב, הפרוייקט אמור לעבור קומפילציה ללא warning ולא error, אך מן הסתם, לא יעבור linker.

את סעיפים ב+ג יש להגיש כקובץ זיפ/RAR לתיקיית ה- google-docs המתאימה עד ה- 26.08.2019 בשעה 23:59. שם הקובץ יהיה כשם הקובץ כפי שתואר בסעיף א'. לפני הגשת קובץ פרוייקט לאתר, יש לעשות clean. קובץ הזיפ צריך להיות עשירות קילו בלבד (לעומת כמה מגה ללא הניקוי והמחיקה).

חלק 2

- א. בחלק זה תבחרו סיפור ומימוש של קבוצה אחרת ותמשיכו מהפתרון שלה. עליכם לממש את כל הקוד החסר כך שהפרוייקט יתקמפל ויעבור linker, ירוץ ויעבוד היטב ללא באגים. מותר לכם בתרגיל זה לשנות במעט את הקוד שעליו אתם מתבססים (c'tor, const וכנראה שבעיקר את ה- main). שימו לב, שתדרשו לענות על שאלות המבוססות על קוד זה ולכן אם אינו כתוב היטב, עליכם לתקנו, אך אין לשנות את המימוש הבסיסי. אם אתם מתלבטים, תשאלו אותי בפורום לגבי אופי השינוי שאתם רוצים לערוך.
- ב. אין להשתמש ב- STL או ב- templates בחלק זה.

חלק 3

- א. ממשו רשימה מקושרת כללית באמצעות templates. יש לתמוך רק בהוספת איבר לסוף, מחיקת איבר והדפסה.
- ב. השתמשו ברשימה שיצרתם בסעיף הקודם על-מנת להחליף את אחד המערכים המוגדרים בקוד.
- ג. החליפו את כל השדות מטיפוס `char*` שהוקצו דינאמית במחלקה `string`
- ד. החליפו את השימוש במערכים לשימוש ב- `vector` של ה-STL (פרט למערך ששיניתם בסעיף ב').
- ה. בעקבות שינוי זה יש למחוק מהפרוייקט את כל השלישיות (`copy c'tor`, `d'tor` ואופרטור השמה) שאינן נחוצות יותר.
- ו. הכניסו לסיפור תוספת כך שיכלול שימוש ב- `design pattern 3`.
- ז. ממשו את ה- `design pattern` שהגדרתם בקוד.

את חלקים 2+3 יש להגיש כל אחד בנפרד כקובץ זיפ/RAR לתיקיות ה-google-docs המתאימה עד ה-22.10.2019 בשעה 23:59. שם הקובץ יהיה כשם הקובץ כפי שתואר בסעיף א' של חלק 1.

מצ"ב בדף הבא רשימת בדיקה עם הדגשים שעליהם הורדו בעבר נקודות בפרוייקט. אין רשימה זו מחייבת שהדגשים יהיו זהים בסמטר זה, אך שווה לא לפספס לפחות בדברים אלו.

עבודה פורייה, מלמדת ונעימה!

דגשים לכתיבת הקוד

const & ו-const

שימוש חלקי ב- const על הפרמטרים
שימוש חלקי ב- const על ערכים מוחזרים
שימוש חלקי ב- const על מתודות
שימוש מיותר ב- const על ערכים המועברים או מוחזרים by value

השלישיה ו- בנייה

מימוש מיותר של השלישיה: copy c'tor, operator=, d'tor
מימוש חלקי של השלישיה: copy c'tor, operator=, d'tor
אין שימוש נכון בהרשאות עבור copy c'tor ואופרטור השמה
הגדרת ושימוש ב- default c'tor
מימוש כמה c'tor'ים במקום שימוש בערכי ברירת מחדל
חסר virtual d'tor
הוספת virtual d'tor לכל המחלקות באופן גורף

הורשה והכלה

יצירת אובייקט חדש כאשר יש ירושה, במקום הפנייה
במימוש השמה לא לקרוא להשמה של האבא
אין קריאה ל- c'tor של אובייקט מוכל בשורת האתחול
הכלה במקום ירושה
הגדרה של פוינטר במקום אובייקט מוכל

חריגות

אין שימוש בחריגות

STL

עבודה עם משתנה המייצג גודל מערך כאשר עובדים עם STL
מעבר על אוספי STL באמצעות אינדקס במקום איטרטור
עבודה עם string כאילו היה טיפוס בסיסי ולא אובייקט (חסרים const ו- &)

הגדרת המערכת

אין מחלקה מנהלת
הקצאת משתנים בהקצאה דינאמית במקום כמשתנים רגילים
מימוש לוגיקה ב- main
יצירת מחלקות מיותרות
הגדרת משתנים גלובליים בתוכנית
הגדרת תכונות ב- public
הגדרת קבועים באמצעות define במקום const
הגדרת enum גלובלי ולא בתוך המחלקה, או שכפול
חסר שחרור זכרון
הגדרת מערך כאשר יש צורך במשתנה יחיד
יש קלט ופלט במחלקות