



המסלול האקדמי

המכללה למינהל

להצלחה יש דרך

תרגיל 7

הקדמה

לעתים נרצה thread pool שמנהל את התור שלו כתור עדיפויות. ה thread pool הרגיל של java.util.concurrent אמנם מאפשר לנו לעבוד עם PriorityQueue אך ה Runnable ו Callable שהוא מקבל אינם ברי השוואה (Comparable) ולכן צריך לבצע אדפטציה מתאימה.

בתרגיל זה עליכם לקרוא עצמאית ולהבין את התיעוד אודות:

- ThreadPoolExecutor (שאותו נירש)
- FutureTask (שיכול לעטוף Runnable או Callable)
- RunnableFuture (שיכול לרוץ כ runnable ולחזור כ future)

וליצור Thread Pool שניתן להזין לו Runnable או Callable יחד עם פרמטר עדיפות שעל פיו הם יצאו מהתור.

בנייה

במחלקה PriorityThreadPool תמצאו מחלקת עזר פנימית ופרטית בשם Helper ואת שתי המתודות שעליכם לממש.

התחילו מכך ש PriorityThreadPool תירש את ThreadPoolExecutor. בעקבות הירושה יהיה עליכם ליצור בנאי מתאים כדי לאתחל את החלק שירשתם. נשים לב שהפרמטר האחרון הוא BlockingQueue<Runnable>. החליפו אותו עם PriorityQueue.

כעת נרצה את האפשרות לעטוף Runnable ו/או Callable באמצעות מחלקה שהיא גם ברת השוואה. לשם כך עליכם לעדכן את המחלקה <V>Helper. חישבו איזו מחלקה עליה לרשת ואלו ממשקים עליה לממש. זכרו שעליה להיות ברת השוואה בצורה שתעדיף ערכי priority נמוכים יותר.

תצטרכו ליצור לה שני בנאים, האחד שעוטף Runnable והשני שעוטף Callable. שניהם יקבלו בנוסף פרמטר int priority ופרמטרים אחרים ע"פ הצורך (קראו בתיעוד).

כעת נתבונן בחתימה של המתודה execute. אנו אמורים לבצע execute לאיזשהו runnable. אך לא נוכל פשוט לקרוא ל super.execute() שירשנו... בהינתן ה priority ניצור אובייקט מתאים מהסוג של Helper. אך super.execute לא מכיר את Helper ולכן יש להסתכל על האובייקט שנוצר דרך "משקפת" אחרת. חישבו איזו.

הערה: יש לנו תלות ב V. אך מכיוון שה Runnable אינו תלוי ב V אז ה JVM תצטרך להסיק מהו V באמצעות פרמטר אחר. לכן יש לנו את הפרמטר V result.

לעומת זאת, במתודה submit אין צורך בפרמטר נוסף משום שה Callable כבר אומר ל JVM מהו V.

המימוש של submit כבר יהיה מאד דומה לזה של execute פרט לשימוש ב Callable וההחזרה של Future מתאים.

בדיקה

כדי להקל עליכם בדיקת ה MainTrain וה MainTest זהות.

- יצרנו Priority Thread Pool חדש שייצר אך ורק ת'רד אחד פעיל.
- הזנו משימה ראשונה שתעכב אותו 3 שניות, כדי לאפשר לשאר המשימות להתמייין בתור ע"פ העדיפות שלהן.
- הזנו 0-49 משימות ריקות מסוג Runnable בעדיפויות 100-51 בהתאמה
- הזנו עוד 50-99 משימות שמחזירות את הערכים 50-99 בהתאמה בעדיפות 1-50 בהתאמה
- נשים לב שמשימות אלה קבלו עדיפויות גבוהות יותר (ערכי priority נמוכים יותר)

- הערה: לצורך דיבאג תוכלו להוסיף הדפסות לערך j בביטויי הלמדה של ה Runnable וה Callable לעיל.
 - ראו את ההערות בסוף ה main כדי לבדוק את הפלט הצפוי
 - כמובן לפני הגשה מחקו הדפסות מיותרות
- הבדיקה היא די פשוטה. בדקנו שכל ה Futures שחזרו אכן מכילים את הערכים 50-99.
- דחפנו משימה אחרונה בעדיפות מאד נמוכה כדי שתפעל אחרונה – הדפסת done.
- וסגרנו את ה thread pool.

עליכם להגיש את PriorityThreadPool.java בלבד ל ex7

בהצלחה