

2. נתונה המחלקה **Patient** – חולה בחדר מיון, ולה שתי תכונות:

- id – מספר הזהות של החולה, מטיפוס שלם
 - priority – רמת הדחיפות של הטיפול בחולה. רמת הדחיפות מיוצגת במספר מטיפוס שלם בין 1 ל-10. ככל שהמספר גבוה יותר, רמת הדחיפות גבוהה יותר.
- הניחו שלתכונות המחלקה יש פעולות `get/Set` ו- `set/Set`.

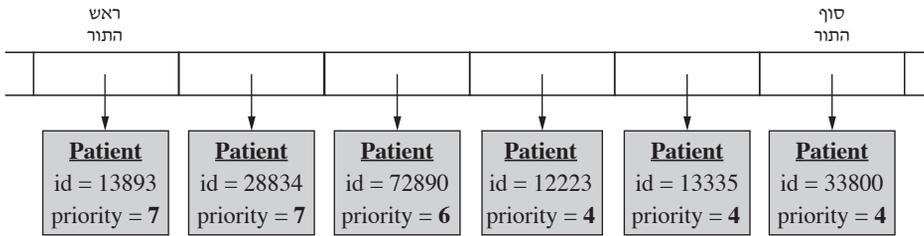
סדר הטיפול בחולים בחדר המיון מתנהל באופן שלהלן:

ככל שרמת הדחיפות של הטיפול גבוהה יותר, החולה מטופל מוקדם יותר. כאשר יש יותר מחולה אחד באותה רמת דחיפות, החולה שהגיע קודם לחדר מיון מטופל מוקדם יותר.

כדי לשמור על סדר הטיפול נבנתה המחלקה **PriorQueue** – תור עדיפויות, ולה תכונה אחת:

- q – הפניה לתור, מטיפוס Patient

דוגמה לתור q:



הניחו שהחולים שרמת הדחיפות שלהם זהה מוצגים בדוגמה לפי סדר הגעתם לחדר המיון.

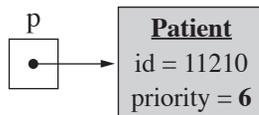
א. ממשו את הפעולה שלפניכם ששייכת לממשק המחלקה `PriorQueue`:

Java – `public void priorityInsert (Patient p)`

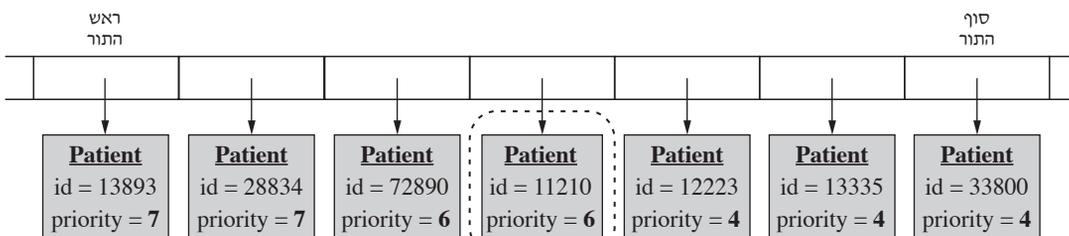
C# – `public void PriorityInsert (Patient p)`

הפעולה מקבלת חולה חדש – p ומכניסה אותו לתור q בהתאם לכללים של חדר המיון הכתובים לעיל.

לדוגמה: עבור התור המוצג לעיל והעצם שלפניכם:



התור ייראה כך לאחר ההכנסה:



ב. מדי פעם רמת הדחיפות של חולה מסוים משתנה במהלך שהותו בחדר המיון.

ממשו את הפעולה שלפניכם השייכת לממשק המחלקה PriorityQueue :

Java – public void update (int id, int pri)

C# – public void Update (int id, int pri)

הפעולה מקבלת מספר זהות של חולה הנמצא בתור – id , ומספר – pri המייצג את רמת הדחיפות המעודכנת שלו. הפעולה תעדכן את התכונה priority של החולה ותמקם אותו בתור לטיפול במקום המתאים לו (בהתאם לרמת הדחיפות המעודכנת – pri). אם בתור כבר יש חולים אחרים באותה רמת דחיפות – pri , החולה הנוכחי – id יוצב אחריהם כאילו הגיע אחריהם לחדר המיון.