

1. אילו דרישות צריכות להתקיים ע"י פרמטרי התבנית Element ו-Compare בהם אנו משתמשים ב-UniqueArray?

הדרישה עבור המחלקה Element היא שיהיה לה בנאי העתקה. אם לא מסופק Compare, אז נדרש גם אופרטור השוואה של Element.

הדרישה עבור המחלקה Compare היא שיהיה לה בנאי חסר פרמטרים, ושיהיה לה אופרטור עם החתימה:

```
bool operator()(const Element&, const Element&);
```

2. נניח כי היינו מגדירים ע"י operator overloading את האופרטור הבא, המקבל מספר סידורי של איבר ב-UniqueArray ומחזיר את המצביע לאיבר:

```
const Element* operator[] (unsigned int index) const;
```

איזו בעיה עלולה להיווצר כעת בשימוש ב-UniqueArray? כיצד ניתן להתגבר על הבעיה?

הבעיה היא שלא יהיה ניתן להשתמש במערך מסוג UniqueArray<unsigned int>, כי אז תהיה התנגשות עם האופרטור המוגדר (const Element&) operator[] – הם יהיו עם אותה חתימה (מלבד זה שאחד מהם הוא עם reference והשני לא), וניסיון לשימוש באופרטור הזה יהיה מעורפל. הקומפיילר לא יוכל לבחור באיזו גרסה של הפונקציה להשתמש ולכן ההידור ייכשל.

על מנת למנוע את הבעיה הזו, נעדיף שלא להעמיס את האופרטור אלא לממש פונקציה נפרדת, למשל:

```
const Element* getElement(unsigned int index) const;
```

ואז לא תהיה התנגשות.

אם בכל זאת אנו נאלצים להשתמש ב-UniqueArray שמומש בו האופרטור כפי שתואר, ונרצה להשתמש בו עם סוג unsigned int, אז נוכל לעטוף את הערכים שלנו במחלקה אחרת כדי להתגבר על הבעיה, למשל:

```
class MyUInt {  
public:  
    unsigned int value;  
    MyUInt(unsigned int v) : value(v) {}  
};
```

וכך נוכל לקבל איבר לפי אינדקס או לפי ערך:

```
UniqueArray<MyUInt> ua(50);  
// ...  
const MyUInt *a = ua[8]; // get by index  
const MyUInt *b = ua[MyUInt(8)]; // get by value
```