

ב"ה

תשובות:

שאלה 1:

הדרישה ההכרחית ביותר שעל הטיפוס לקיים היא אופרטור > תקף בין אובייקטים מהטיפוס. אחרת, לא ניתן לסדר את הרשימה בסדר הנדרש

שאלה 2:

גישה לשינוי הטיפוס על ידי האיטרטור, תביא לשבירת הסדר ברשימה מבחינת האופרטור > של הטיפוס. כי יהיה ניתן לשנות את גודל אובייקט הטיפוס ישירות, ולשבור את הסדר ברשימה

שאלה 3:

על ידי הגדרת מבנה ליצירת אובייקט פונקציה בוליאנית, פרדיקט.

שדות: אינטגר של המחלק

בנאי: עם פרמטר אינטגר במקבל את ערך המחלק

מתודה: בוליאנית המחזירה אמת אם משתנה מטיפוס שלם כלשהו שמפעיל אותה מתחלק בשלם בשדה של אובייקט הפונקציה

כעת, במתודה שנרצה להשתמש בפילטר, נבנה / נגדיר אובייקט של המבנה פרדיקט שיצרנו, ובארגומנט הבנאי נשלח את המחלק שמתקבל בזמן הריצה. לאחר שבנינו את האובייקט, כשנשתמש בפילטר נשים בארגומנט את שם האובייקט פרדיקט שיצרנו, וזהו.

הערה: יש גם עוד אפשרויות

הצעת מימוש (נקודתית וחלקית):

מחלקה:

```
Class Numbers {
    SortedList<int> m_numbers;
    struct IsDivisedBy;
public:
    void onlyDivisedBy (unsigned int divider);
}
```

מבנה יצירת אובייקט הפרדיקט:

```
struct Numbers::IsDividedBy {  
    unsigned int m_divider;  
    explicit IsDividedBy(unsigned int divider) : m_divider(divider) {}  
    bool operator()(int number) const {  
        return number % divider == 0;  
    }  
};
```

שימוש בפרדיקט:

```
Numbers::onlyDividedBy(unsigned int divider) {  
    // Build the functor object for run time argument value divider  
    IsDividedBy filterFunctor(divider);  
    // Use the filter function to create a list of numbers divisible by the divider  
    m_numbers = numbers.filter(filterFunctor);  
}
```