ObjectEditor.NET

אפיון וניתוח

קורן בר

גרסה 1.0

תוכן עניינים

שם הפרויקט: ObjectEditor.NET
תחום הפרויקט: כלי דינאמי לעריכת נתונים מבוסס על עקרונות של תכנות מונחה עצמים.
מטרה כללית: יצירת כלי המאפשר למשתמשים לערוך כל אובייקט נתונים בצורה דינאמית, באמצעות חקר המבנה הפנימי של האובייקט ושימוש ביכולות Reflection.

מבוא

בפרויקטים רבים, מתכנתים נדרשים לבנות טפסים אינטראקטיביים לעריכת נתונים. לרוב, יש ליצור ממשק משתמש מותאם למודל הנתונים, מה שמוביל לעבודה כפולה עם תחזוקה מסורבלת.
ObjectEditor.NET היא ספרייה דינאמית שמטרתה לספק פתרון גנרי לעריכת כל אובייקט בכל רמת מורכבות, תוך שמירה על גמישות ואינטואיטיביות. הספרייה מבוססת על ארכיטקטורת MVC ומנצלת את היכולות של Reflection לביצוע תהליכים דינאמיים.

ייעוד המערכת

ObjectEditor.NET מיועדת בעיקר למתכנתים ומהנדסי תוכנה הזקוקים לכלי עריכה גנרי לתמיכה בתהליכי פיתוח. המערכת:

- חוסכת את הצורך ביצירת View ייעודי לכל מודל.
- מתאימה לעבודה עם מבנים מורכבים כמו אובייקטים מקוננים, רשימות, ומילונים.
- מאפשרת הגדרת מגבלות גישה או עריכה באמצעות Attributes בלבד, ללא צורך בקוד נוסף.
- משפרת את חוויית המשתמש עם ממשק גרפי אינטואיטיבי שמשקף ישירות את מבנה האובייקט.

תיאור הפרויקט

ObjectEditor.NET מבוססת על חלוקה ברורה לשכבות (MVC):

1. **Model**:
   - מייצג את הנתונים והמבנה של האובייקט לעריכה.
   - נתמך על ידי Reflection לקריאה וכתיבה דינאמית של מאפיינים.
   - כולל מנגנון הרשאות וגישה באמצעות Attributes.
2. **View**:
   - ממשק משתמש המבוסס על WinForms.
   - מספק תצוגה אינטראקטיבית של המודל, כולל תמיכה באובייקטים מקוננים.
   - כולל ווידג'טים מותאמים לעריכת ערכים בהתאם לסוג המידע.
3. **Controller**:
   - אחראי לתקשורת בין ה-Model ל-View.
   - כולל שני קונטרולרים מרכזיים:
     - `ObjectEditorController` לניהול האובייקט כולו.
     - `FieldController` לניהול מאפיינים ספציפיים.

תכונות עיקריות:

- קריאה דינאמית של אובייקטים ותכונותיהם.
- עריכת ערכים בזמן אמת עם עדכון ממשק המשתמש.
- תמיכה באובייקטים מקוננים (Nested Objects) באמצעות גישה רקורסיבית.
- מנגנון הרשאות מובנה לעריכת נתונים.

דרישות המערכת

דרישות פונקציונליות:

1. המערכת תאפשר עריכת ערכים של כל מאפיין באובייקט נתון.
2. תמיכה בעבודה עם רשימות, מערכים ואובייקטים מקוננים.
3. הצגת כל המאפיינים, כולל סוג המידע, שמות המאפיינים, וערכיהם הנוכחיים.
4. ממשק משתמש גרפי (GUI) עם אפשרות לעריכה אינטראקטיבית.

דרישות ביצועים:

1. תגובה תוך פחות מ-500ms לכל פעולה בממשק המשתמש (למעט טעינת הדף).
2. תמיכה בעבודה עם לפחות 10,000 אובייקטים בו-זמנית.
3. צריכת זיכרון סבירה שתאפשר ריצה גם במחשבים מוגבלי משאבים.

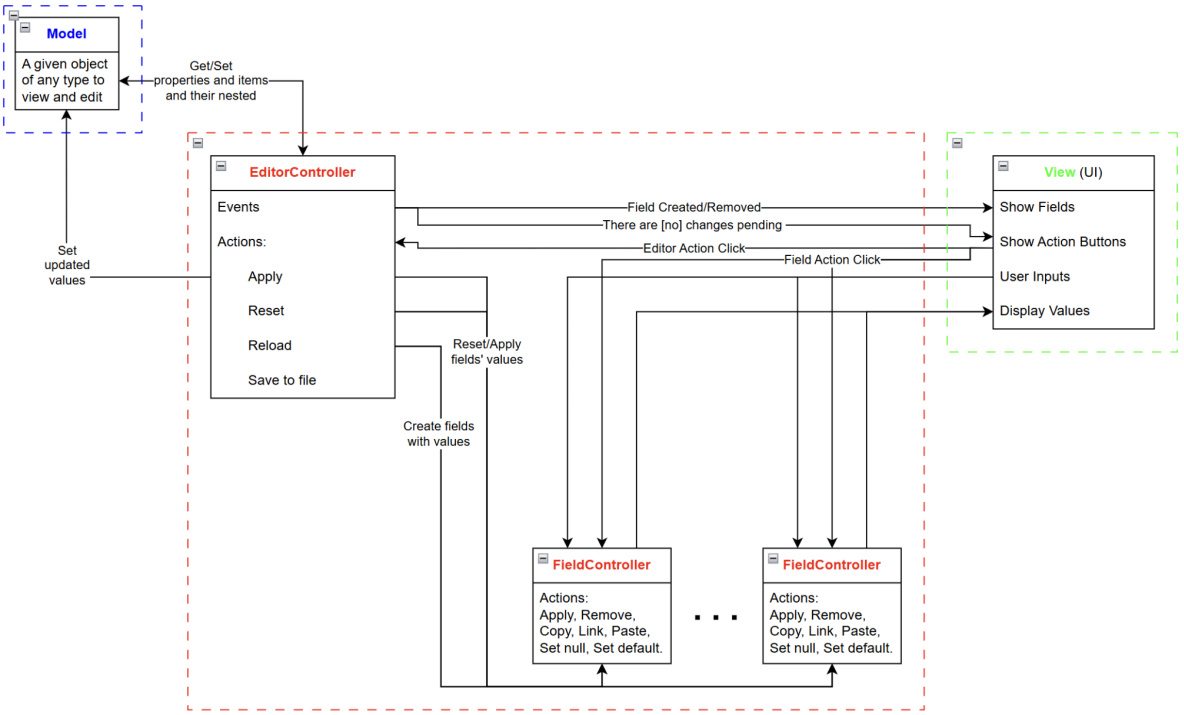
דרישות אבטחה:

1. מנגנון למניעת שגיאות עקב עריכת נתונים לא חוקיים.
2. טיפול בשגיאות (Exception Handling) למניעת קריסות של המערכת.
3. מנגנון מובנה לאימות ערכים לפני שמירתם.
4. הגבלת גישה לפי קבוצות הרשאות.
5. מניעת גישה לא חוקית למאפיינים מוסתרים.

מבנה המערכת

מבנה ה-MVC בפרויקט:

- **Model**:
  - המידע נאסף בעזרת `Reflection`.
  - מאפיינים ניתנים לעריכה מוצגים עם הסוג הנכון (למשל `int`, `string`).
  - תמיכה בערכים מקוננים באמצעות גישה רקורסיבית.
  - תמיכה במערכים ורשימות עבור כל סוג של `T` גנרי.
- **View**:
  - בנוי ב-WinForms וכולל גריד דינאמי לתצוגת המידע.
  - חלוקה לקטגוריות עבור מאפיינים מורכבים.
  - תמיכה בעריכת ערכים עם ווידג'טים (controls) מותאמים לסוג הנתון (תיבת טקסט, תיבת סימון וכו').
- **Controller**:
  - מתווך בין `View` ל-`Model`.
  - מנהל את פעולות הקריאה והכתיבה.
  - מטפל ביוצאים מן הכלל במקרים של גישה לא חוקית או סוגי נתונים לא תואמים.



תרחישי שימוש (Use Cases)

1. עריכת נתון פשוט

תרחיש: המשתמש בוחר באובייקט נתון ומבצע עדכון לערך של אחד המאפיינים.
שלבים:

1. בחירת אובייקט בממשק המשתמש.
2. איתור המאפיין הרלוונטי.
3. עדכון הערך באמצעות תיבת טקסט.
4. אישור השינויים ועדכון האובייקט המקורי.

2. גישה לערכי אובייקט מקונן

תרחיש: המשתמש עורך מאפיין בתוך אובייקט שמקונן באובייקט הראשי.
שלבים:

1. פתיחת רשימת המאפיינים של האובייקט הראשי.
2. איתור המאפיין המקונן.
3. לחיצה לפתיחת רשימת המאפיינים שלו.
4. בחירת המאפיין הפנימי לעריכה.
5. שמירת השינויים ועדכון האובייקט.

3. העתקה, הדבקה, קישור, איפוס ערך, או מחיקת פריט ברשימה

תרחיש: המשתמש מבצע פעולה על מאפיין או פריט באובייקט.
שלבים:

1. איתור המאפיין באמצעות אחד מהתרחישים הקודמים.
2. לחיצה ימנית בעכבר להצגת תפריט הפעולות.
3. לחיצה על הפעולה הרצויה על ערך המאפיין.

דגשים טכנולוגיים

1. **Reflection**: שיטה דינאמית לעבודה עם סוגי אובייקטים לא ידועים מראש.
2. **Attributes**: מנגנון קל לשימוש להגדרת מגבלות גישה על מאפיינים.
3. **MVC**: עיצוב מונחה אחריות שמבטיח גמישות ותחזוקה קלה.
4. מנגנון רקורסיבי לקריאה וכתיבה של ערכים במאפיינים מקוננים.
5. תמיכה ברשימות ומילונים עבור כל טיפוס נתונים גנרי.

הנחות עבודה והגבלות

1. כל האובייקטים הנתונים חייבים להיות נגישים באמצעות Reflection.
2. המערכת אינה תומכת בסוגי נתונים מותאמים אישית שאינם ממומשים בצורה נכונה.
3. כרגע אין תמיכה בשמירת השינויים לקובץ חיצוני (Future Scope).

תרחישים עתידיים להרחבת המערכת

1. הוספת אפשרות לייצוא וייבוא נתונים בפורמטים כמו JSON או XML.
2. פיתוח מנגנון הרשאות מפורט יותר לעריכת מאפיינים.
3. הוספת תצוגת גרף (Graph View) למיפוי אובייקטים מורכבים.
4. מימוש ה View ב Web Application עבור שימוש .