

# Simulation eines Fernüberwachungssystems für Photovoltaikanlagen

## 5. und 6. Praktikumstermin, Y-Gruppe

*Praktika sind Prüfungsvorleistungen, die zum genannten Termin erbracht werden müssen. Das Programm (C++-Projekt) wird während des Praktikums zum angegebenen Zeitpunkt im Informatik-Labor abgenommen.*

*Zu Beginn dieses Praktikumstermins müssen Sie Kenntnisse zur Thematik Klassen, Objekte, Vererbung, grafische Oberflächen und Ereignisbehandlung nachweisen und den prinzipiellen Lösungsweg erläutern können.*

## Ziel

Die Echtzeit-Daten einer Sammlung von Photovoltaikanlagen werden von zentralen Stellen eingesammelt und als ständig aktualisierte Grafiken dargestellt. Diese Aufgabe soll über zwei Praktikumstermine in Teams zu zwei oder drei Personen bearbeitet werden. Für Teams aus drei Personen ist die Aufgabenstellung umfangreicher (siehe unten).

## Architektur

**Tipp: Zeichnen Sie die Klassenhierarchie als UML-Klassendiagramm!**

Es gibt zwei abstrakte Basisklassen:

- `TDatenanbieter` beschreibt die grundlegenden Funktionen, mit denen Daten abgerufen werden können:

```
class TDatenanbieter
{
    public:
    virtual double getLaenge() = 0;
    virtual double getBreite() = 0;
    virtual double getLeistung() = 0; // Watt
    virtual double getStrahlung() = 0; // Watt pro Quadratmeter
    virtual double getPaneltemperatur() = 0; // Kelvin
};
```

- `TPhotovoltaikanlage` stellt eine Photovoltaikanlage dar. Hiervon erben verschiedene spezielle Typen an Photovoltaikanlagen. Die Klasse `TPhotovoltaikanlage` selbst erbt von `TDatenanbieter`.

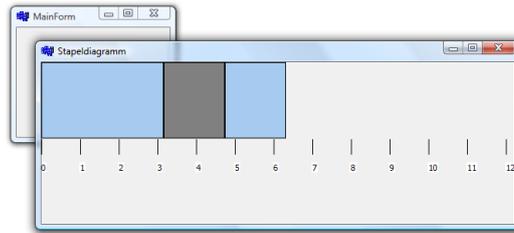


Abbildung 1: Das Diagramm stellt die von links nach rechts gestapelten Leistungsdaten dar.

Die Klasse `TDatensammler` hilft, Datenquellen zu verwalten. Sie besitzt eine Funktion, mit der man `TDatenanbieter` bei ihr anmelden kann, durch Übergabe eines Zeigers auf den `TDatenanbieter`. Diese Anmeldefunktion kann mehrfach aufgerufen werden, um mehrere `TDatenanbieter` zu registrieren. `TDatensammler` speichert diese intern mit `push_back` in einem `std::vector<TDatenanbieter*>` (recherchieren Sie gegebenenfalls, wie `std::vector` anzuwenden ist). Außerdem besitzt diese Klasse eine Elementfunktion, die zurückgibt, wie viele `TDatenanbieter` aktuell angemeldet sind, sowie eine Elementfunktion, die den Zeiger auf den  $n$ -ten `TDatenanbieter` zurückgibt.

Das Programm wird als VCL-Fomularanwendung angelegt. Das Fenster des Hauptformulars bleibt leer. Es enthält einen `TDatensammler`, die `TPhotovoltaikanlagen` und eine oder mehrere Zeiger auf Anzeigefenster als private Datenelemente. Im Konstruktor des Hauptformulars werden die `TPhotovoltaikanlagen` beim `TDatensammler` angemeldet und das oder die Anzeigefenster mit `new` angelegt und mit ihrer Elementfunktion `Show` geöffnet.

Eine `TPhotovoltaikanlage` kennt ihre Position (geographische Länge, Breite), ihre aktuelle Leistung, die einfallende Strahlung und die Paneltemperatur. Es sollen zwei Kindklassen von `TPhotovoltaikanlage` implementiert werden, die bei Abfragen auf jeweils verschiedene Arten (zum Beispiel mit der logistischen Gleichung, vgl. Mathematik-Vorlesung) eine pseudzufällige Globalstrahlung  $W$  und Paneltemperatur liefern (zur Simulation, mangels realer Messwerte). Die Leistung schätzen Sie aus  $W$  mit einem üblichen Wirkungsgrad und einem leichten Abfall mit steigender Paneltemperatur, abgelesen z. B. aus realen Kennlinien (Link).

Das Anzeigefenster wird im C++-Builder als Formular erzeugt. Es enthält einen privaten Zeiger auf den `TDatensammler`. Der Konstruktor des Anzeigefensters ist so zu erweitern, dass dieser Zeiger zusätzlich übergeben wird.

In regelmäßigen Zeitintervallen (Ereignis `OnTimer` eines `TTimer`-Objekts, siehe Hilfefunktion) ruft das Anzeigefenster seine Funktion `Invalidate` auf, um ein Neuzeichnen für den eigenen Fensterinhalt anzufordern. Die Ereignisbehandlungsroutine von `OnPaint` holt dann vom `TDatensammler` die Zeiger auf die `TDatenanbieter` und ruft darüber die Funktionen zum Holen der Daten auf. Das Zeichnen geschieht mit den Elementfunktionen der Eigenschaft `Canvas` des Formulars.

Es soll ein Anzeigefenster implementiert werden, das die aktuellen Energieerträge der Anlagen von links nach rechts gestapelt zeigt, siehe Abbildung 1. Zeichnen Sie eine Kilowatt-Skala an das Diagramm (50 Pixel = 1 Kilowatt).

**Drei-Personen-Teams** sollen eine weiteres Anzeigefenster implementieren, welches die aktuellen Strahlungen und Paneltemperaturen an den Anlagen mit Hilfe von mehr oder minder großen gelben Kreisscheiben und roten Thermometerbalken auf einer Landkarte zeigt. Zum Bestimmen der Längen und Breiten an den Eckpunkten einer Landkarte können Sie Google Maps verwenden (Link). Interpolieren Sie dazwischen linear.