CORDI

Presented by D.Rafał, I.Rudolf

CORDI STANDS FOR

COMPREHENSIVE

Holistyczne podejście do bezpieczeństwa.

ORGANIZATIONAL

Dostosowane do potrzeb instytucjonalnych.

RISK

Proaktywne identyfikowanie zagrożeń.

DETECTION

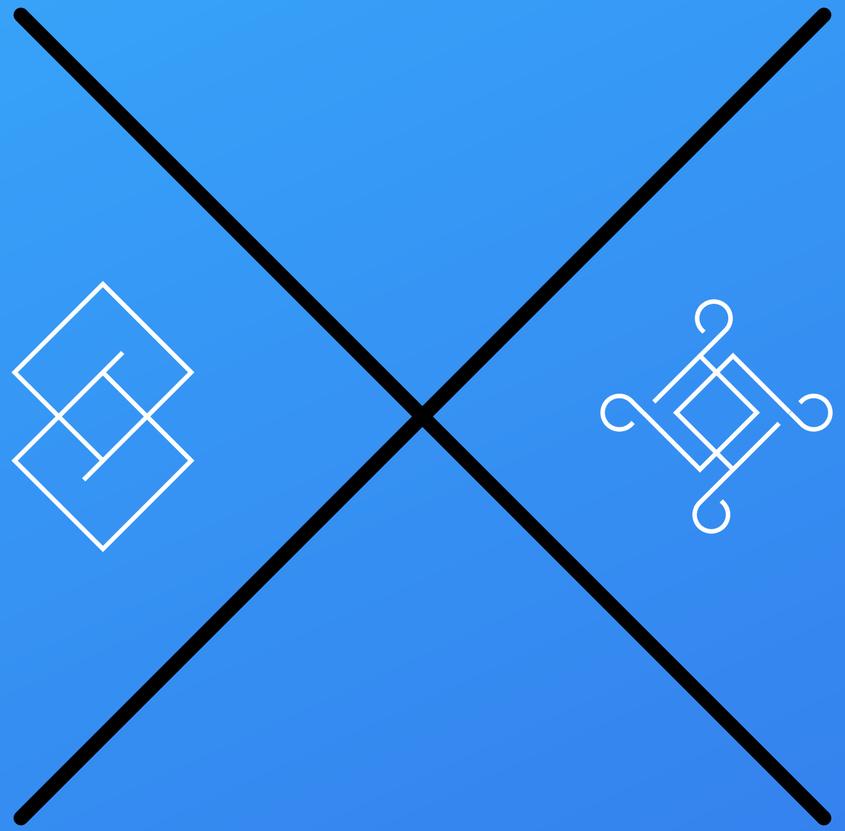
Precyzja w wykrywaniu,

INTELLIGENCE

Najlepsi specjaliści stojący za rozwiązaniami.

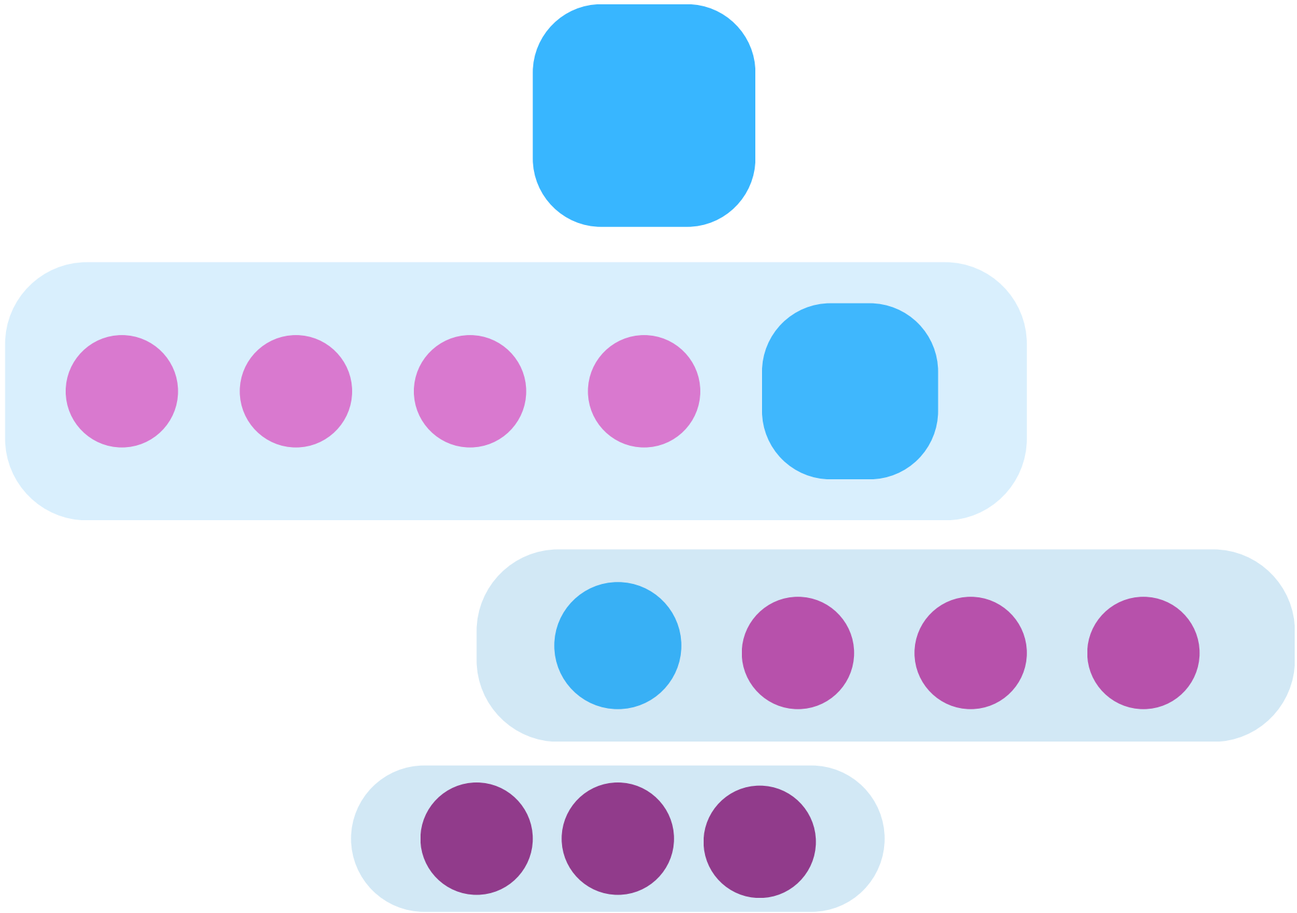
UŻYTECZNOŚĆ DLA DWÓCH ŚWIATÓW

Inteligentnie proste raporty oraz wnikliwe statystyki



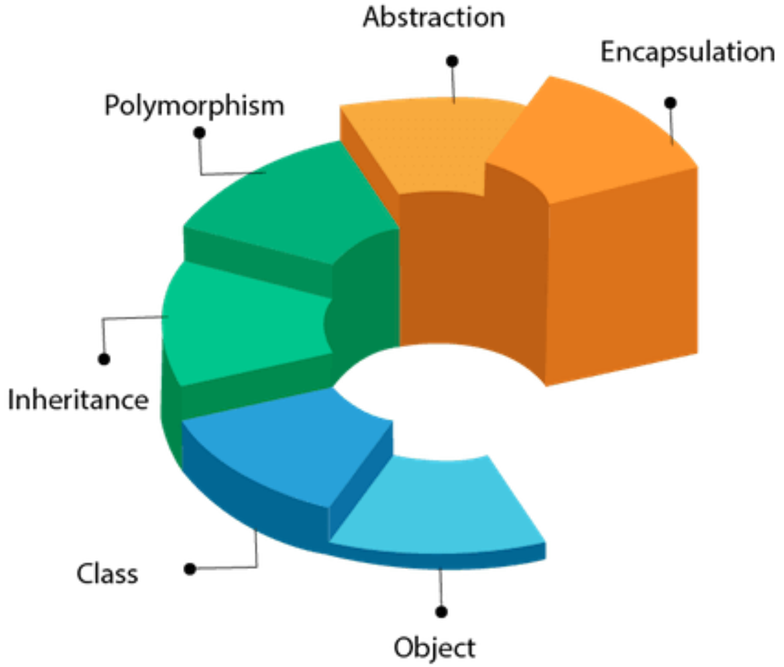
```
from cordi import Cordi
```

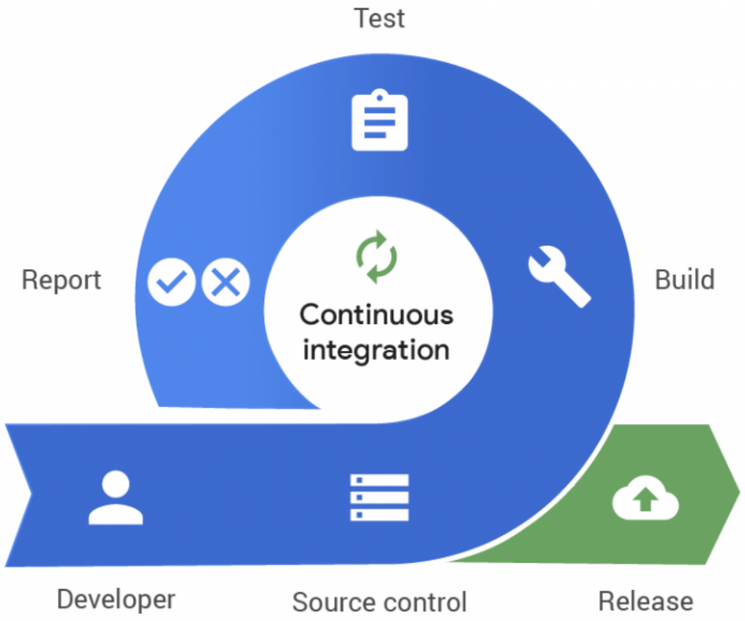
```
cordi=Cordi(..//bhl_folder).
```



CORDI patrzy na całą strukturę plików oraz zagnieżdżone foldery

OOPs (Object-Oriented Programming System)












CSV

Z szacunkiem do języków i prostoty.

```
cordi=Cordi(..//bhl_folder, len="fr").
```

bo francuzi nie umieją w angielski...

File content analysis

Diamond inheritance problem

• BaseClass is a common base class for [**CorrectSubclass**, **SubclassWithoutSuper**]

• Employee is a common base class for [**Developer**, **Manager**]

Not using the super() function

• The Employee class inherits from [**ABC**] and does not use super() in the constructor.

• The Developer class inherits from [**Employee**] and does not use super() in the constructor.

• The Manager class inherits from [**Employee**] and does not use super() in the constructor.

• The Mammal class inherits from [**Animal**] and does not use super() in the constructor.

• The Primate class inherits from [**Mammal**] and does not use super() in the constructor.

• The Human class inherits from [**Primate**] and does not use super() in the constructor.

• The Person class inherits from [**Human**] and does not use super() in the constructor.

• The SubclassWithoutSuper class inherits from [**BaseClass**] and does not use super() in the constructor.

File content analysis

Diamond inheritance problem

• BaseClass est une classe de base commune pour [**CorrectSubclass**, **SubclassWithoutSuper**]

• **Lemployé est une classe de base commune pour** [Développeur, Gestionnaire]

Not using the super() function

• La classe Employee hérite de [**ABC**] et n**utilise pas super() dans le constructeur.**

• La classe Developer hérite de [**Employee**] et n**utilise pas super() dans le constructeur.**

• La classe Manager hérite de [**Employee**] et n**utilise pas super() dans le constructeur.**

• La classe Mammifère hérite de [**Animal**] et n**utilise pas super() dans le constructeur.**

• La classe Primate hérite de [**Mammal**] et n**utilise pas super() dans le constructeur.**

• La classe Human hérite de [**Primate**] et n**utilise pas super() dans le constructeur.**

• La classe Person hérite de [**Human**] et n**utilise pas super() dans le constructeur.**

• La classe SubclassWithoutSuper hérite de [**BaseClass**] et n**utilise pas super() dans le constructeur.**

ale umieją we francuski

“Usus magister est optimus”~Cicero

Not using the super() function

• La classe Employee hérite de [**ABC**] et n**utilise pas super() dans le constructeur.**

• La classe Developer hérite de [**Employee**] et n**utilise pas super() dans le constructeur.**

• La classe Manager hérite de [**Employee**] et n**utilise pas super() dans le constructeur.**

• La classe Mammifère hérite de [**Animal**] et n**utilise pas super() dans le constructeur.**

• La classe Primate hérite de [**Mammal**] et n**utilise pas super() dans le constructeur.**

• La classe Human hérite de [**Primate**] et n**utilise pas super() dans le constructeur.**

• La classe Person hérite de [**Human**] et n**utilise pas super() dans le constructeur.**

• La classe SubclassWithoutSuper hérite de [**BaseClass**] et n**utilise pas super() dans le constructeur.**

Solutions

• To resolve diamond inheritance involving **BaseClass**, ensure all derived classes use cooperative multiple inheritance with **super()**. This approach ensures each class in the hierarchy is initialized exactly once. Below is an example of how to correctly implement constructors in derived classes: class CorrectSubclass(BaseClass): def __init__(self, *args, **kwargs): super().__init__(*args, **kwargs) # Correctly initialize parent classes # Initialize any additional attributes for this class here class SubclassWithoutSuper(BaseClass): def __init__(self, *args, **kwargs): super().__init__(*args, **kwargs) # Correctly initialize parent classes # Initialize any additional attributes for this class here Additionally, consider redesigning the class hierarchy to reduce complexity or ambiguity, potentially eliminating the need for multiple inheritance.

to ty masz wybór:

to ty masz wybór:

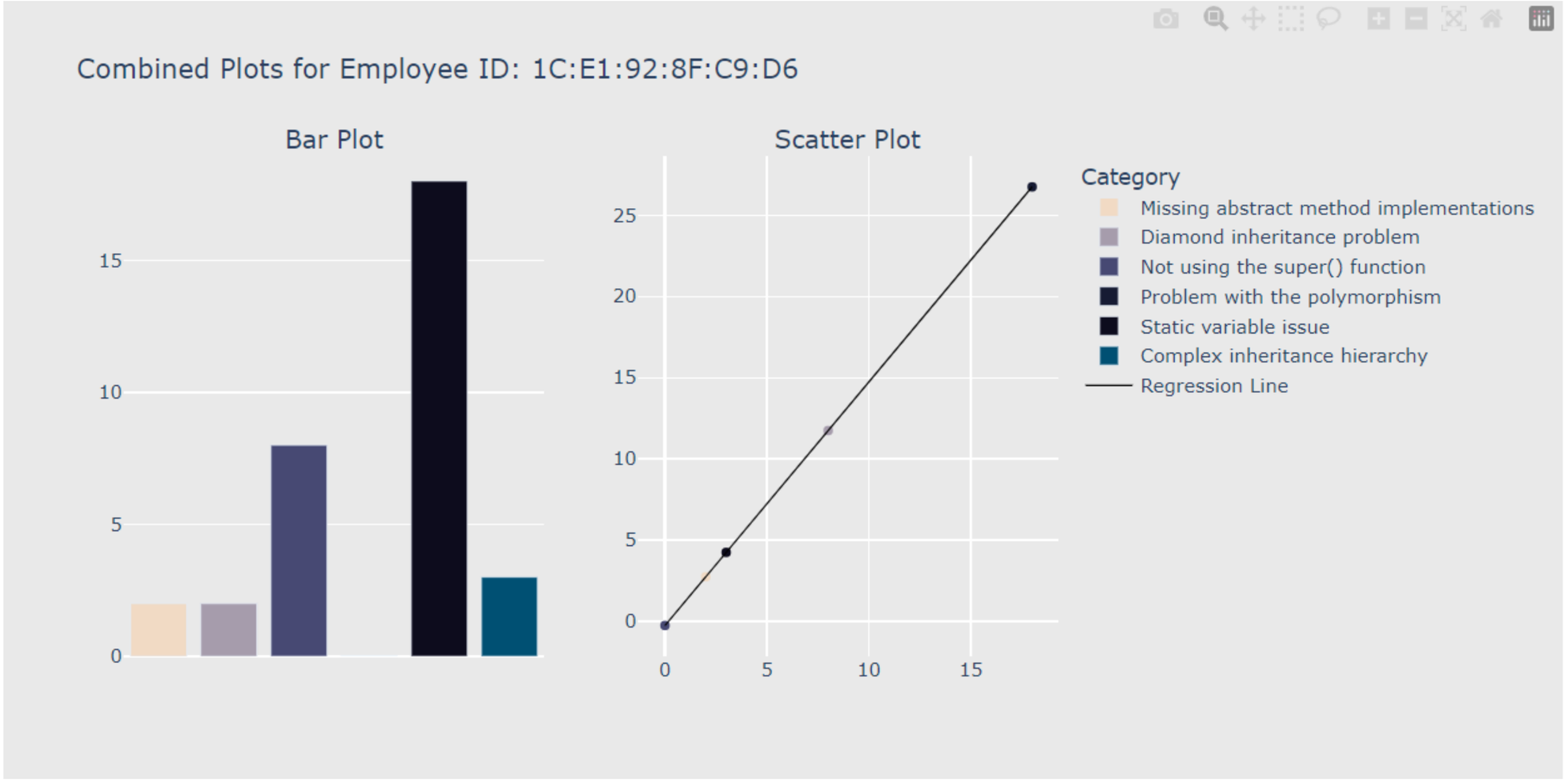
co przetwarzasz?

jak przetwarzasz?

w najprostszej formie

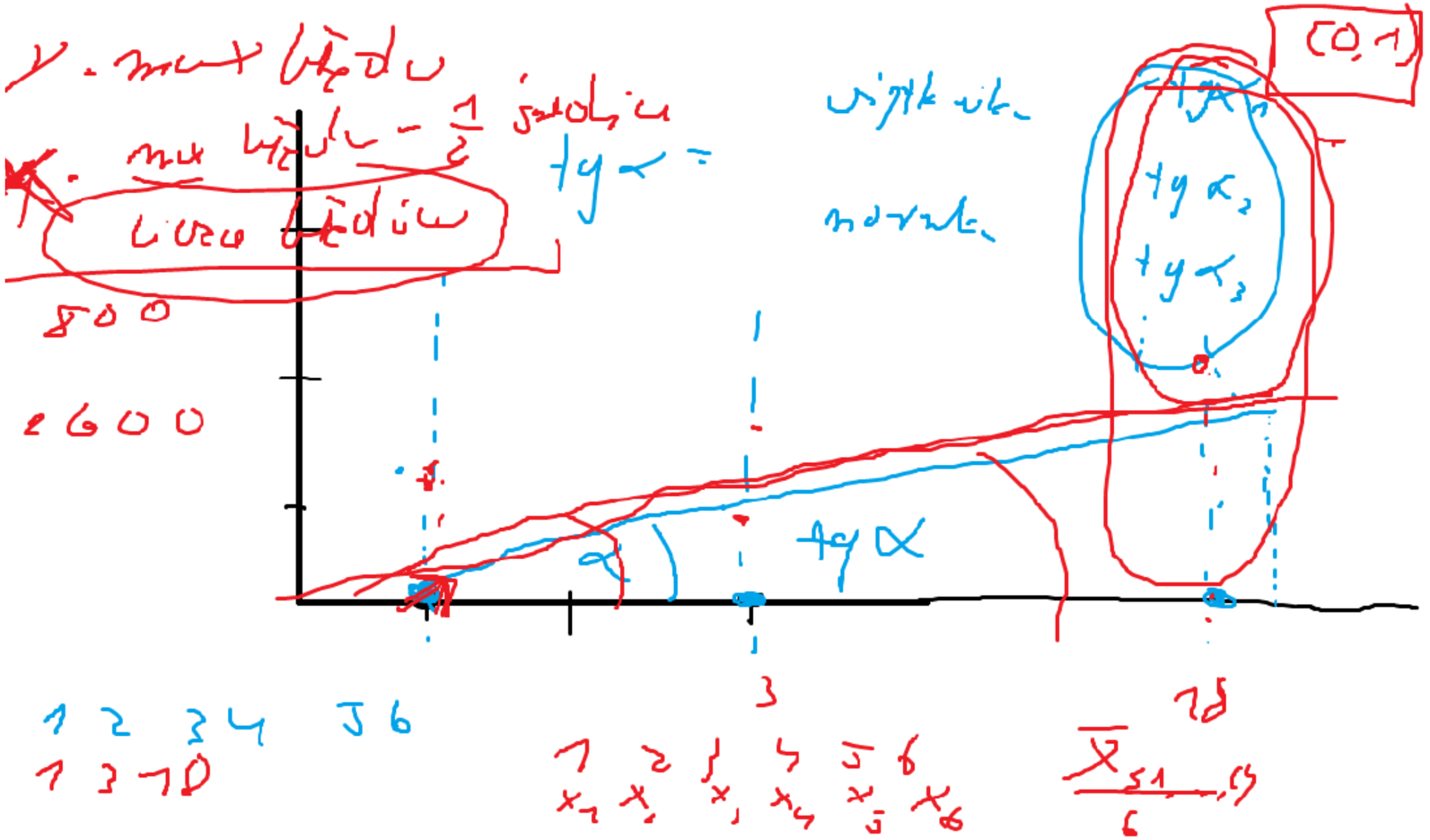
w dowolnej formie

```
cordi=Cordi(..//bhl_folder, len="fr", r_type="basic").

cordi=Cordi(..//bhl_folder, len="fr", r_type="plots").

cordi=Cordi(..//bhl_folder, len="fr", r_type="combined"").
```


Combined Plots for Employee ID: 1C:E1:92:8F:C9:D6
Bar Plot
0
5
10
15
Scatter Plot
0
5
10
15
20
25
0
5
10
15
Category
Missing abstract method implementations
Diamond inheritance problem
Not using the super() function
Problem with the polymorphism
Static variable issue
Complex inheritance hierarchy
Regression Line

“autorska” ocena użyteczności pracowników

za zdjęcie przepraszamy...



DZIĘKUJEMY

ZA POŚWIĘCONY

CZAS