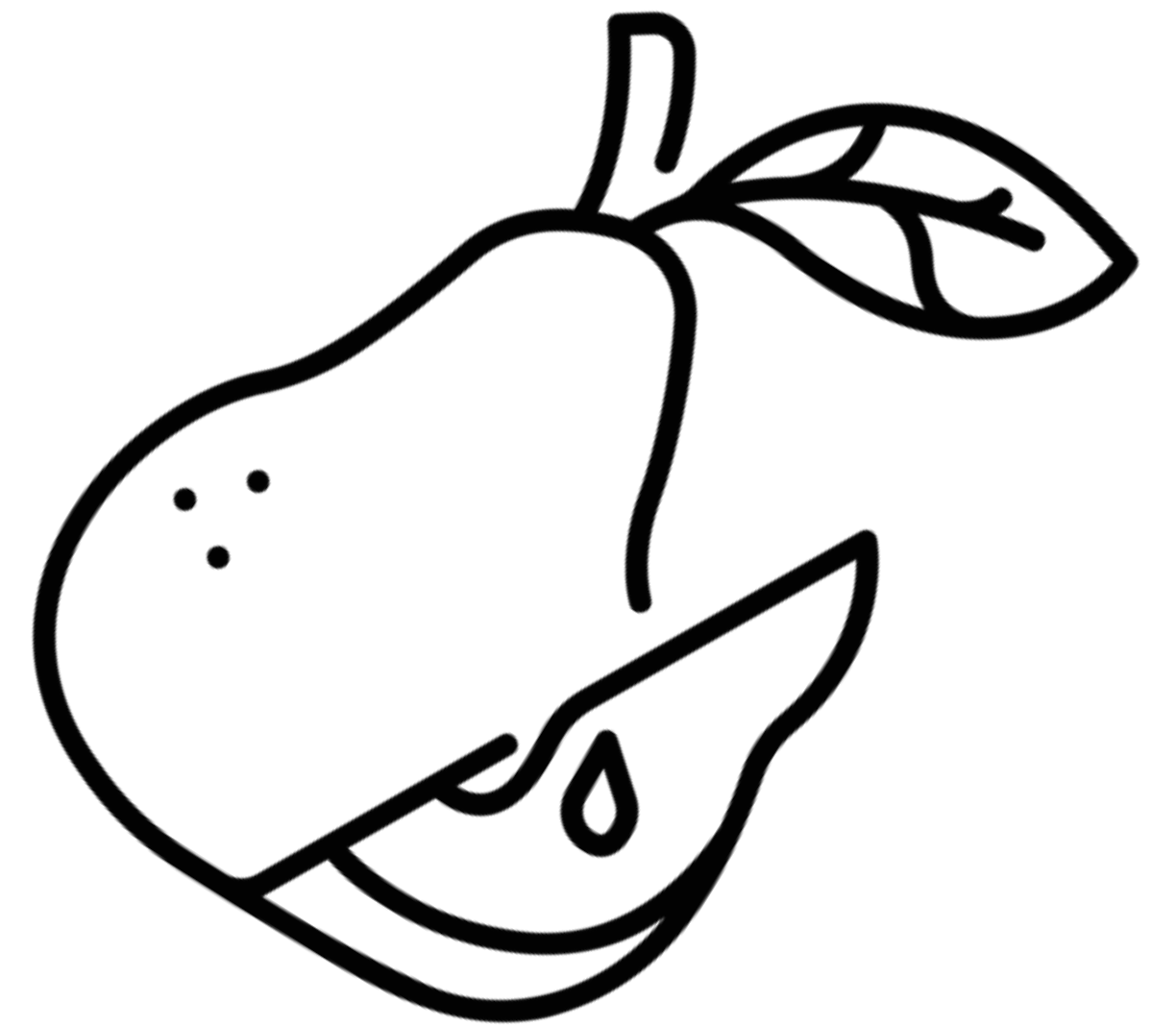
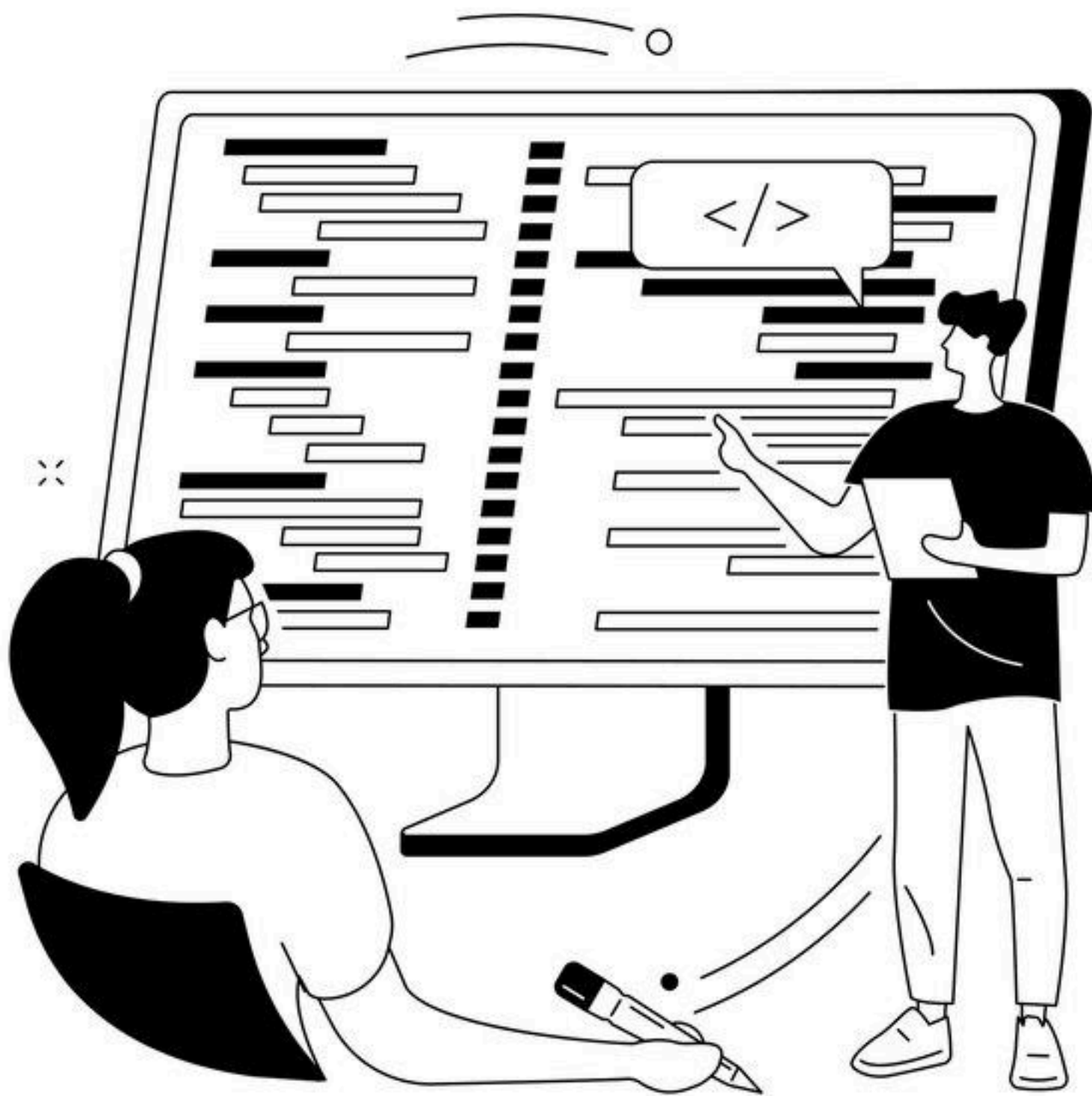


# Мова TotalPear

Вознюк Назар Геннадійович  
Михайлович Олександр Дмитрович  
Група: ТВ-22





# Зміст

01 Короткий опис

02 Приклад синтаксису

03 Особливості

04 Підсумок

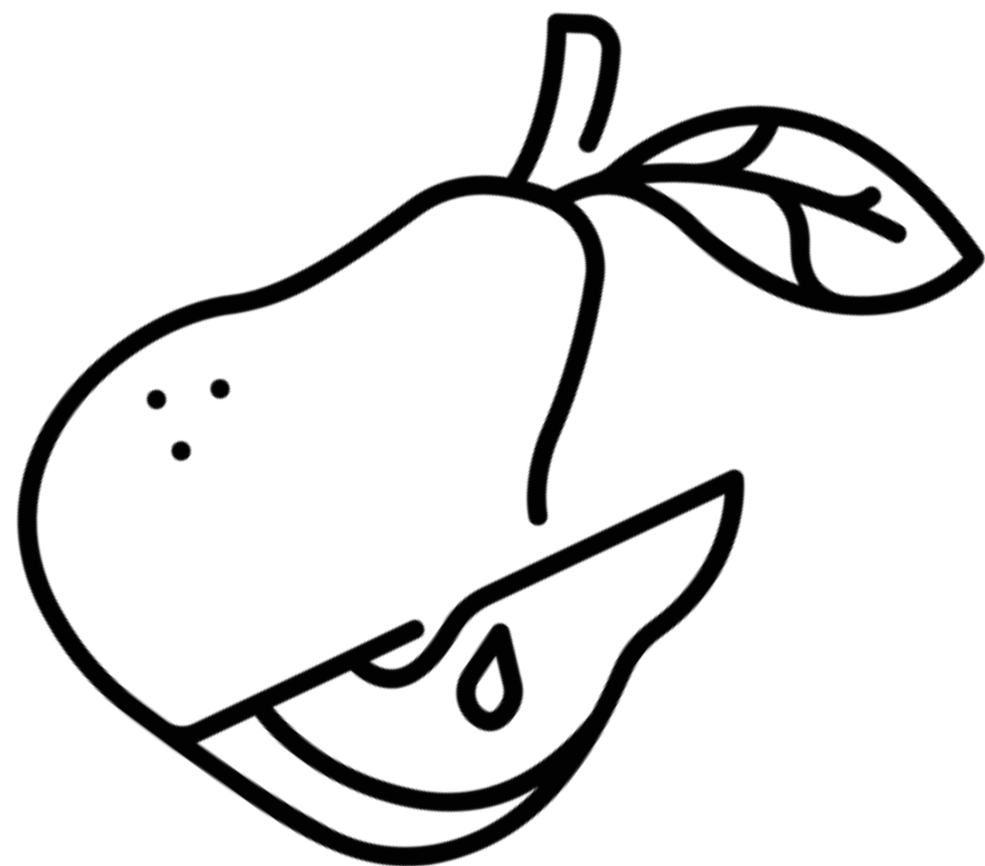
# Короткий опис



Прототипом до розробки цієї мови стала мультипарадигмова мова Dart.



Dart



Назва мови вимовляється як “ті пі”. Назва є аббревіатурою до повної назви мови TotalPear.

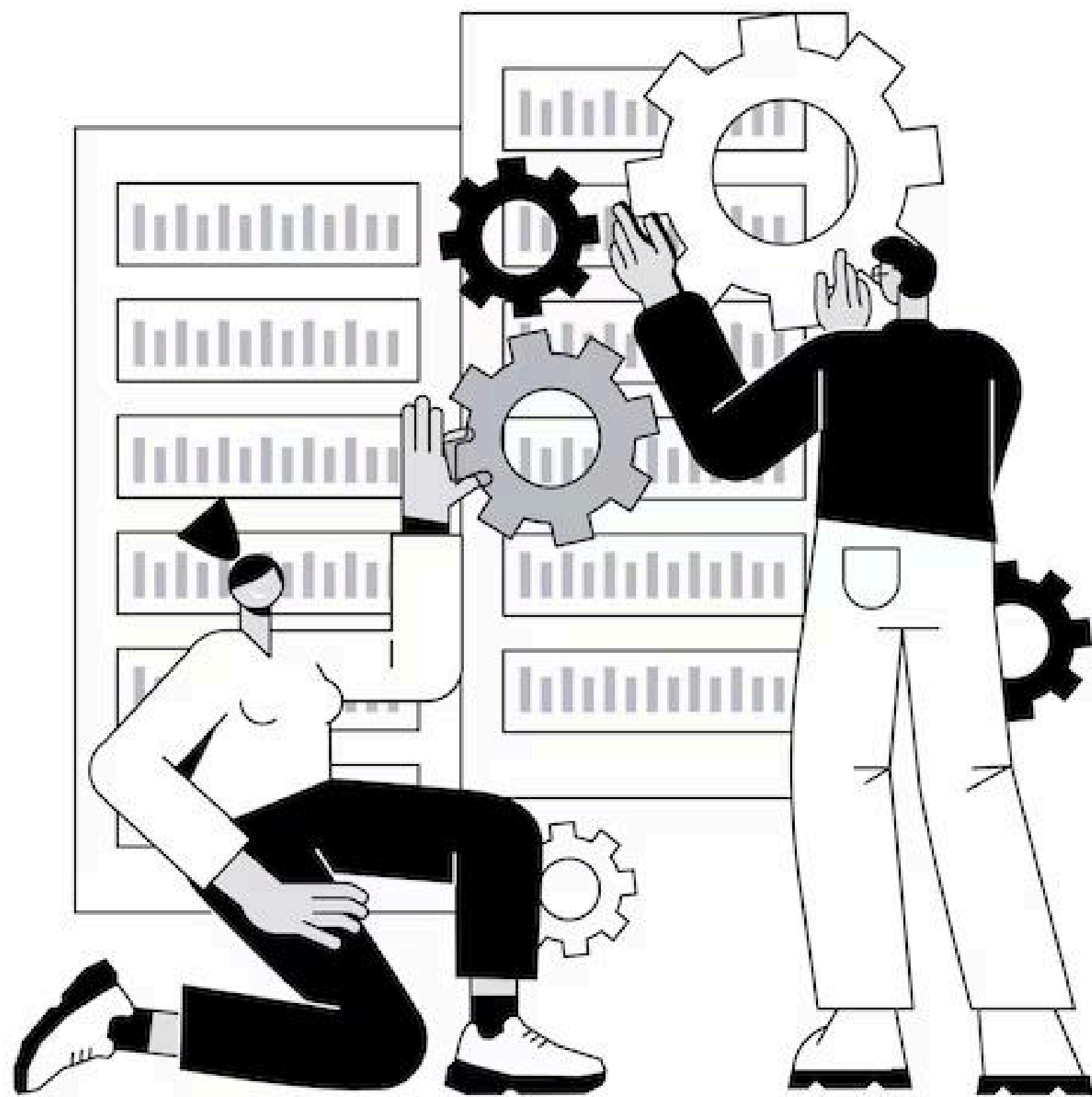
Програма складається з списку інструкцій та підтримує роботу з цілими і дійсними числами та логічними виразами.

```
int x; //testing of default value
int y;

do {
    y = x ^ 2; // first time y is expected to be 0
    print("x: ", x, " | ", "y: ", y, "\n");
    x = x + 1;
} while |y < 144|;
print("Out of cycle with x = ", x, "\n");
```

Для роботи з числами доступні  
4 основні арифметичні операції,  
а також піднесення до степеня  
та круглі дужки.  
Також у мові використовується  
унарний мінус.





**Логічні вирази підтримують  
такі операції, як логічні ‘і’, ‘або’  
та заперечення.  
Також в мові TR доступні  
операції порівняння.**

# Приклад синтаксису

```
print("====Factorial evaluation program\n====");
print("x! = ?\n");
int x = inputInt("Enter x: ");
while |x < 0| {
    print("Incorrect input, x (", x, ") should be ≥ than 0\n");
    x = inputInt("Try again, enter x: ");
}
int i;
int result = 1;
for (i = 1; |i ≤ x|; i = i + 1) {
    result = result * i;
}
print("\nResult: ", result, "\n");
```

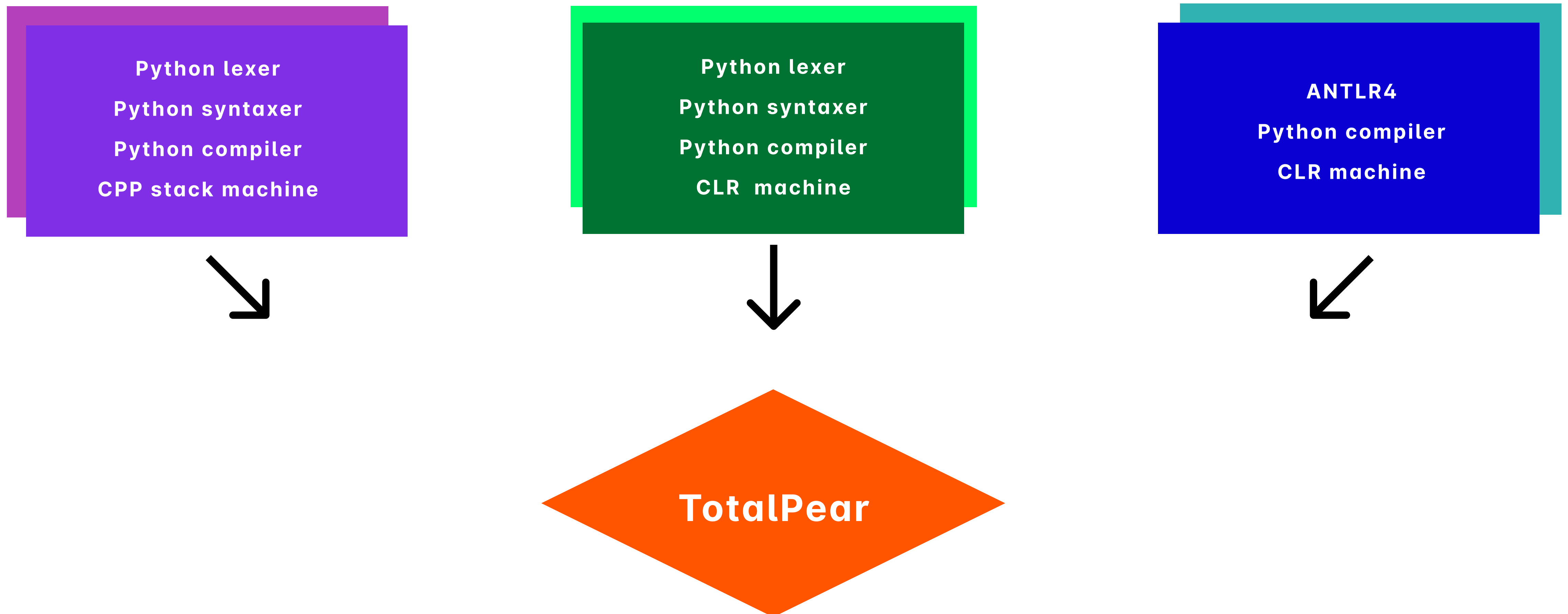






# Особливості

# Системы для запуска



# flagIf

flagIf [#flag1: 'logical1', #flag2: ...]

#flag1: {}

#flag2: {}

...

// Evaluate flag clauses -> bool map

#flag1: {} // on

#flag2: {} // off

... // ...

```
> Input integer: 12.2  
> Error try again: 12.2  
> Error try again: ...
```

```
> Input float: /help  
> Error try again: rm -rf /  
> Error try again: ...
```

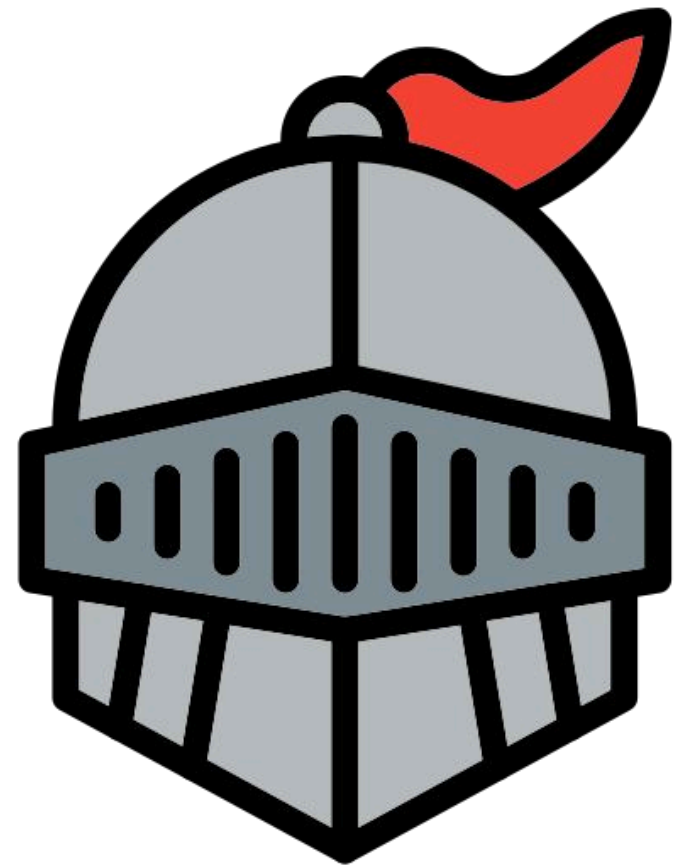
```
> Input bool: yes  
> Nice next one: +  
> Error try again: ...
```



# Persistent inputs

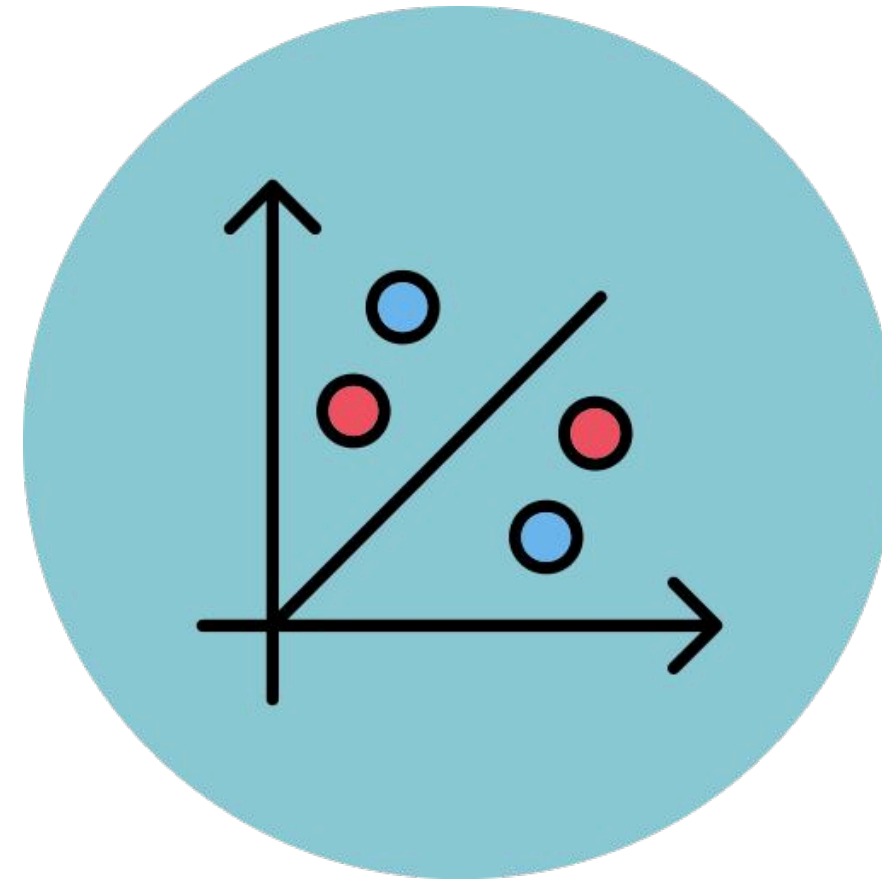


# Підсумок



## Імперативна

Має чітку структуру синтаксису з використанням змінних, присвоєнь, умов і циклів



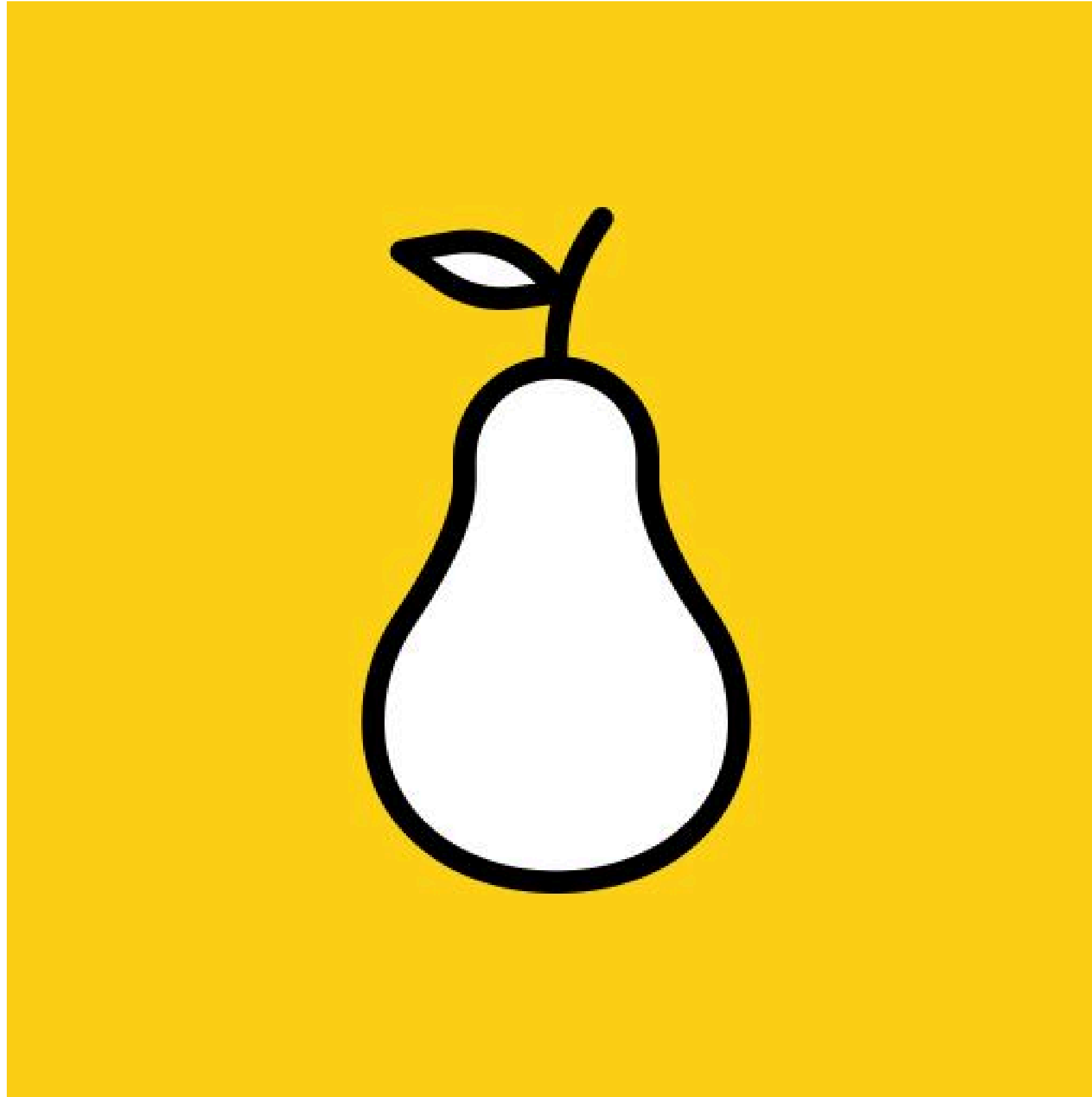
## Типізована

Мова TotalPear має статичну типізацію і підтримує операції з такими типами як int, float та bool. Також присутнє неявне приведення типів int до float та навпаки.



## Декілька систем запуску

Мова TotalPear підтримує запуск як на власній стековій машині так і на CLR. Також є можливість трансляції в CIL як власними силами так і з використанням ANTLR4



Total  ear