



Programowanie komputerów I

Wyjątki

Wątki

Willy Picard

Katedra Technologii Informacyjnych
Akademia Ekonomiczna w Poznaniu
<picard@kti.ae.poznan.pl>

Agenda

- ▶ Cel(e) wykładu
- ▶ Odświeżenie i przekąski
- ▶ Wyjątki
- ▶ Wątki
- ▶ Podsumowanie

Cel(e) wykładu



Przegląd wykładu

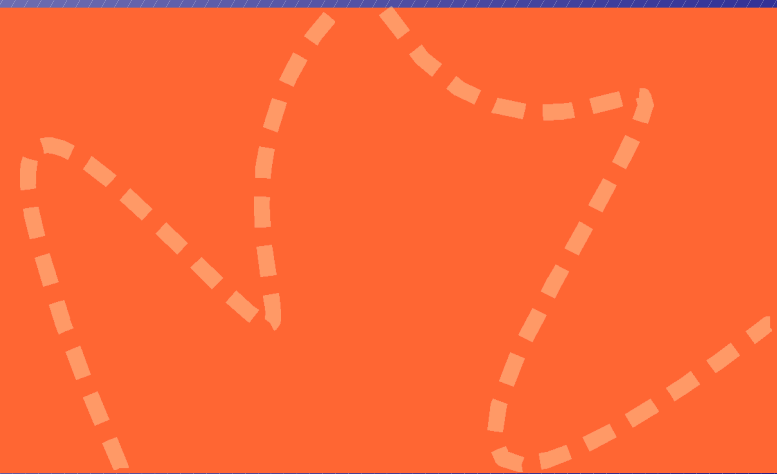
Podstawowe pojęcia

- ▶ 1: Wprowadzenie
- ▶ 2: Podstawowe struktury danych & instrukcje
- ▶ 3: Programowanie obiektowe I
- ▶ 4: Programowanie obiektowe II
- ▶ 5: Programowanie obiektowe III
- ▶ 6: Zaawansowane struktury danych
- ▶ 7: Wątki & Wyjątki

Cel na dziś

Wprowadzić
wyjątki i wątki

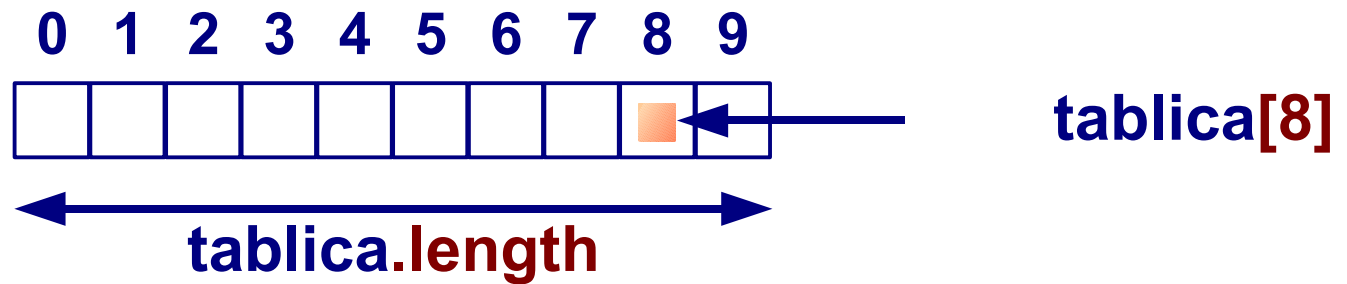
Odświeżenie i przekąski



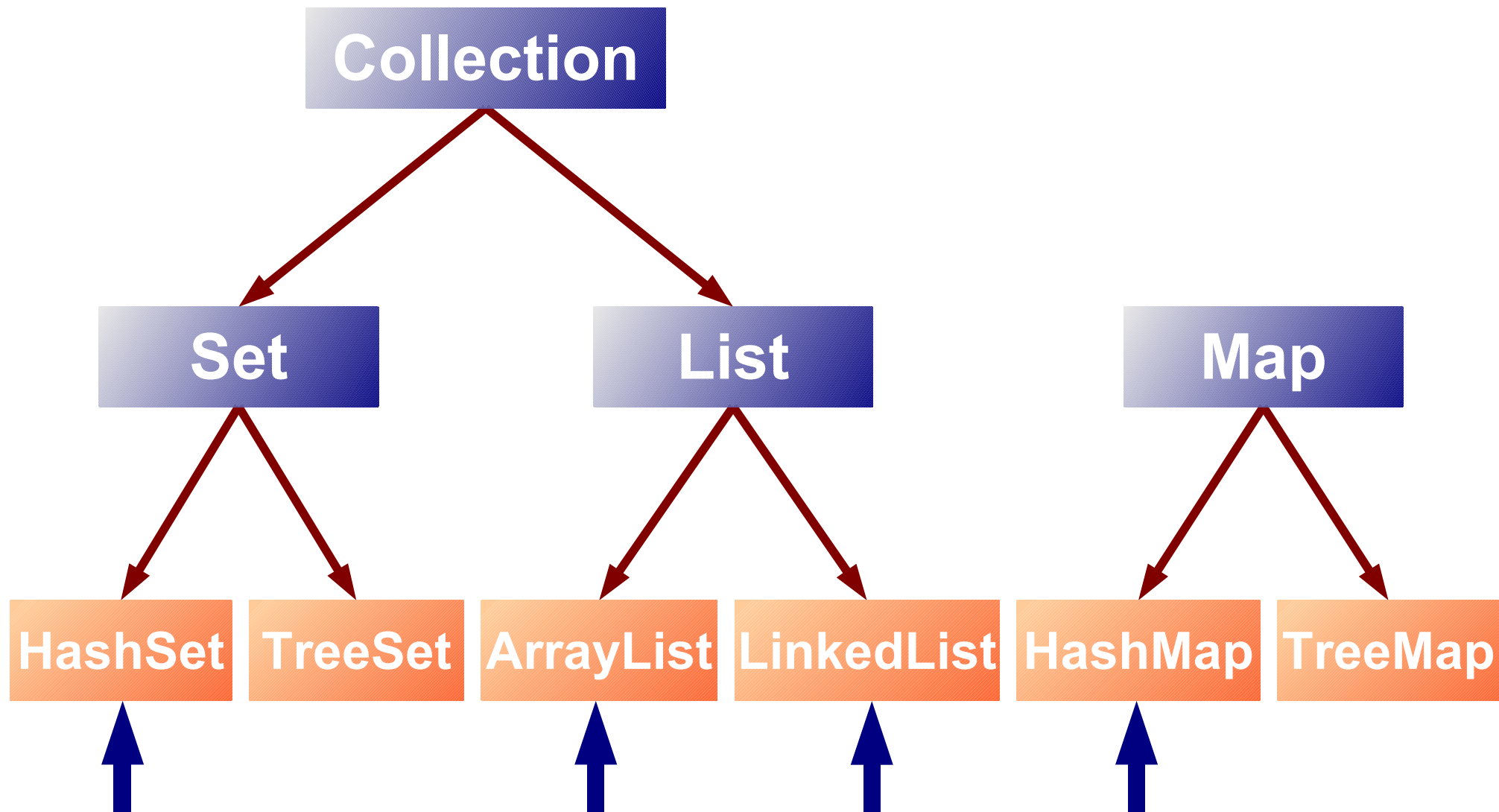
The API Specification

- ▶ Dokumentacja
 - ▶ Pakietów
 - ▶ Interfejsów
 - ▶ Klas
 - ▶ Dziedziczenia
 - ▶ Atrybutów
 - ▶ Metod

Tablice w Javie



Implementacje



Wyjątki



Ariane 5



- ▶ 4 czerwca, 1996, pierwszy lot
 - ▶ 40 sekund
 - ▶ Straty 500\$ milionów
- ▶ Powód
 - ▶ Błąd w oprogramowaniu
 - ▶ Zdarzyło się coś **wyjątkowego**

Definicja wyjątku

Wyjątek jest zdarzeniem w trakcie działania programu, które uniemożliwia programowi dalsze normalne funkcjonowanie

Przykład wyjątku w Javie

```
List koty = new ArrayList();  
koty.add(new Kot("Felix", 1200));  
Cat trzeciKot = koty.get(3);
```



- ▶ IndexOutOfBoundsException
- ▶ **Java API !!!**

Definicja bloku obsługi wyjątku

Blok obsługi wyjątku jest
blokiem kodu
uruchomionym jako reakcja
na wystąpienie wyjątku
danego typu

try-catch-finally w Javie

► Składnia

```
try {  
    <instrukcje>  
} catch (TypWyjatk1 wyjatek1) {  
    <instrukcje>  
} catch (TypWyjatk2 wyjatek2) {  
    <instrukcje>  
} finally {  
    <instrukcje>  
}
```

Przykład *try-catch-finally*

```
List koty = new ArrayList();
koty.add(new Kot("Felix", 1200));
Cat trzeciKot;
try {
    trzeciKot = koty.get(3);
} catch (IndexOutOfBoundsException e) {
    System.out.println("Znamy tylko " +
        koty.size() + " koty");
    System.err.println(e.getMessage());
}
```


Metody i wyjątki

► Składnia

```
► <definicjaMetody> throws <typWyjątku>{  
    ...  
}
```

► Przykład

```
► public void jedz(int ilość)  
    throws Exception {  
    ...  
}
```

Rzucenie wyjątku

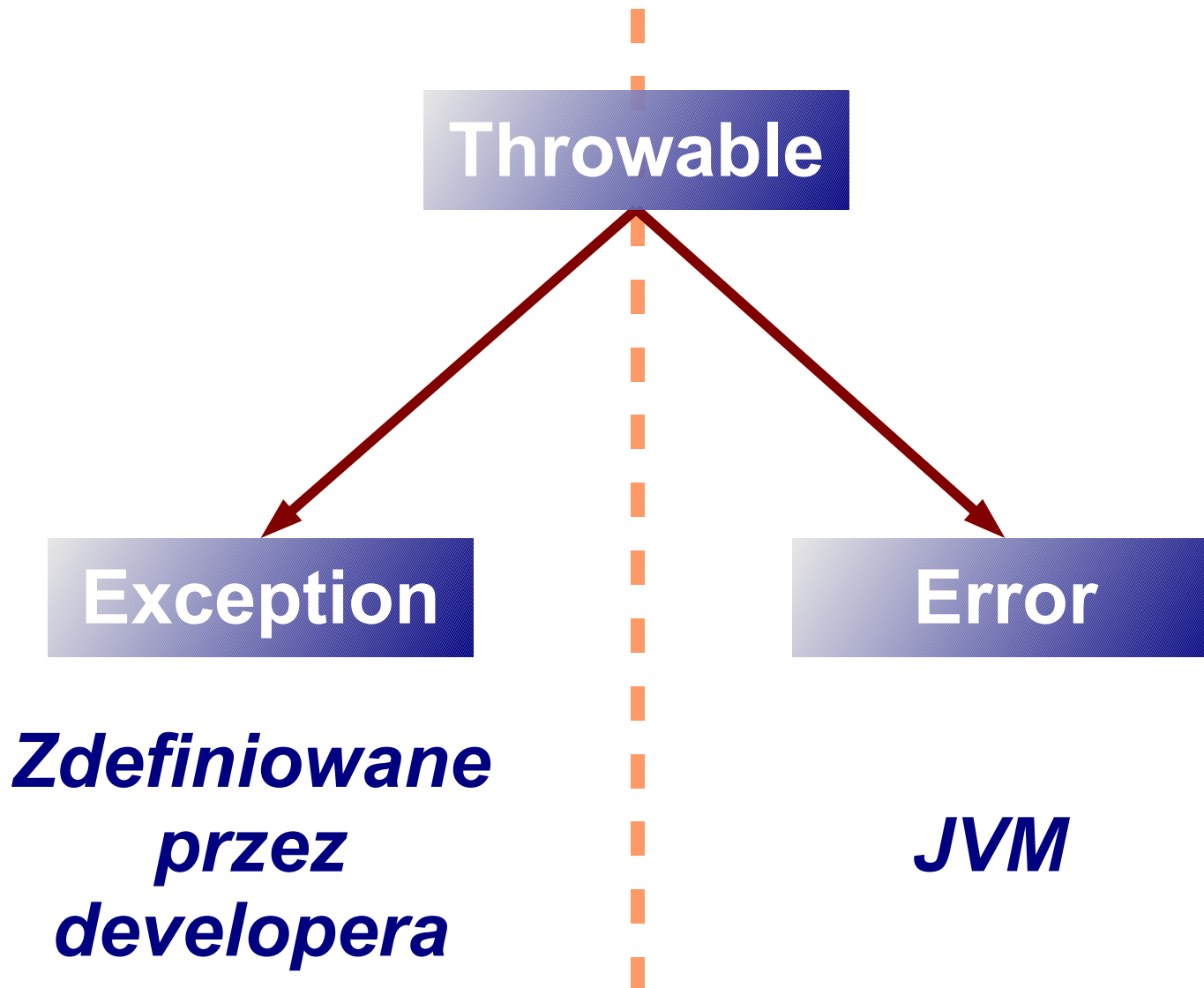
► Składnia

```
► <definicjaMetody> throws <typWyjątku>{  
    ...  
    throw new <typWyjątku>();  
}
```

► Przykład

```
► public void jedz(int ilość)  
    throws Exception {  
    if (ilość <= 0) {  
        throw new Exception("Tylko dodanie"+  
            "ilości karmy!");  
    }  
    ...  
}
```

Exception vs Error



Definicja własnego typu wyjątku

► Składnia

```
► class <nazwaWyjątku> extends Exception{  
    ...  
}
```

...

```
► throw new <nazwaWyjątku>();
```

Definicja własnego typu wyjątku

► Przykład

```
public class NiepoprawnaIlośćWyjatek
    extends Exception{
    private int _ilość;

    public NiepoprawnaIlośćWyjatek(int ilość) {
        super("Ilość karmy "+ilość+
            " jest niepoprawna");
        _ilość = ilość;
    }

    public int zwróćIlość() {
        return _ilość;
    }
}
```

Przykład wyjątku

Wyjątkowe zwierzęta

Wątki



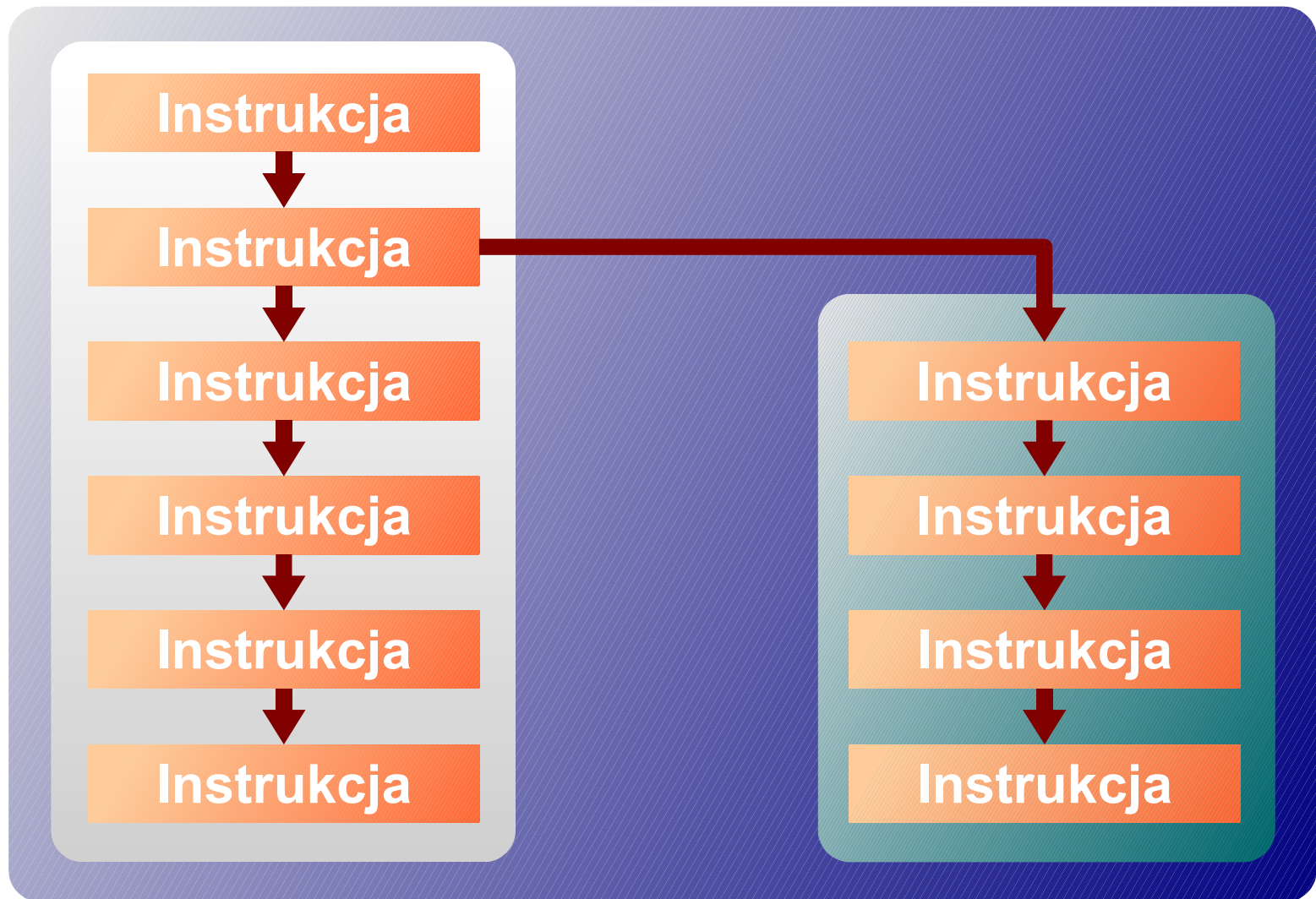
Wielowątkowość

- ▶ Program
 - ▶ Ciąg zadań
 - LUB
 - ▶ Wiele ciągów zadań

Definicja wątku

Wątek jest ciągiem zadań
w ramach jednego
programu

Wielowątkowy program



Wielowątkowość w Javie

- ▶ Dwie możliwości
 - ▶ Rozszerzenie `java.lang.Thread`
 - ▶ Implementacja `java.lang.Runnable`

Rozszerzenie *java.lang.Thread*

► Przykład definicji nowego wątku

```
public class MójWątek extends Thread{  
    public void run() {  
        ...  
    }  
}
```

► Przykład uruchomienia nowego wątku

```
...  
Thread mójWątek = new MójWątek();  
mójWątek.start();  
...
```

Implementacja *java.lang.Runnable*

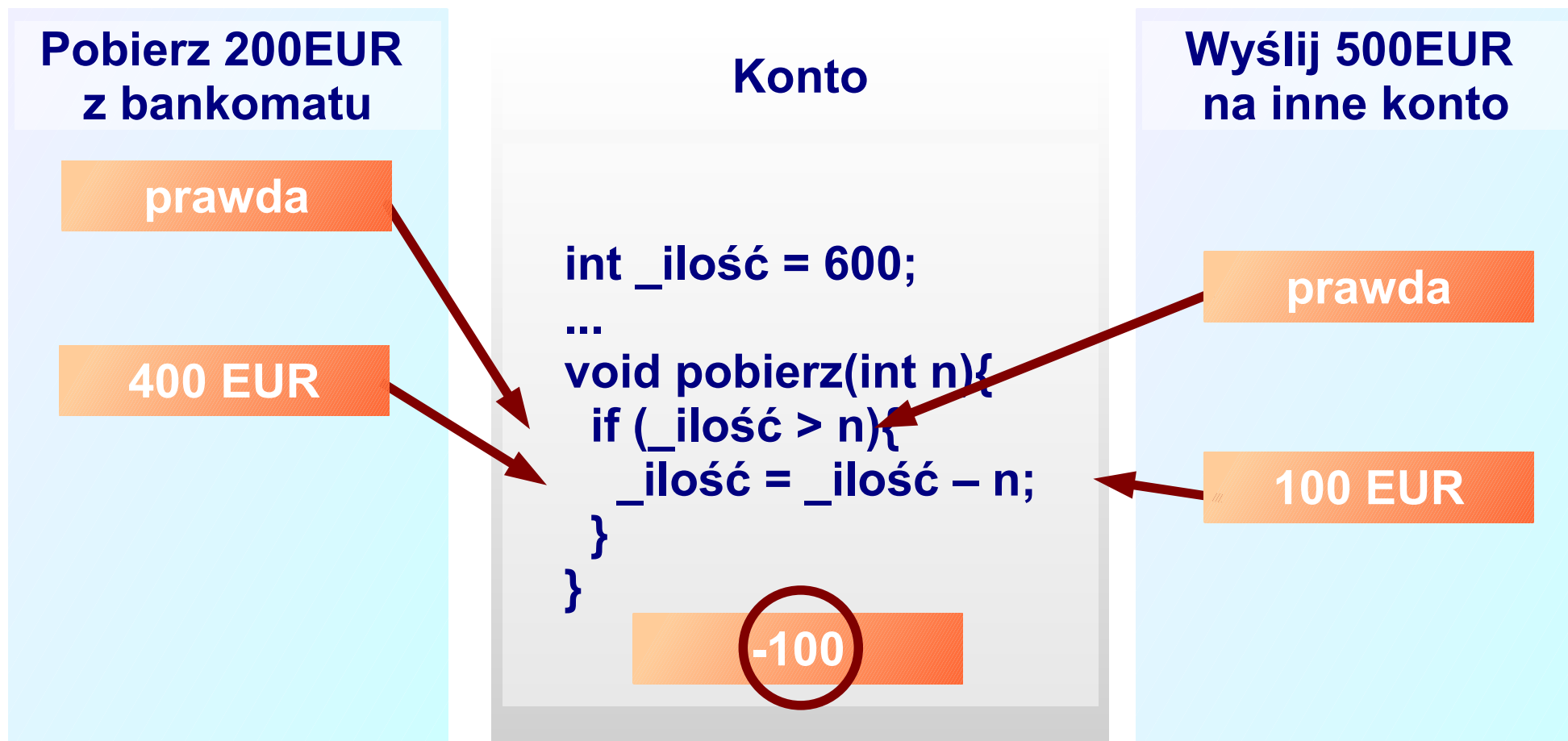
► Przykład definicji nowego wątku

```
public class MójWątek implements Runnable {  
    public void run() {  
        ...  
    }  
}
```

► Przykład uruchomienia nowego wątku

```
...  
Runnable mójWątek = new MójWątek();  
Thread innyWątek = new Thread(mójWątek);  
innyWątek.start();  
...
```

Współbieżność



Słowo kluczowe *synchronized*

- ▶ Blokada
 - ▶ Metody
 - ▶ Ciągu instrukcji
 - ▶ Bloku
- ▶ Tylko jeden wątek posiada blokadę
- ▶ Blokada związana z jednym obiektem

Przykłady *synchronized*

- ▶ Przykład zablokowanej metody

```
public synchronized void pobierz(int n) {  
    ...  
}
```

- ▶ Przykład zablokowanych instrukcji

```
public void pobierz(int n) {  
    ...  
    synchronized(obiekt) {  
        ...  
    }  
}
```


Współbieżność

Pobierz 200EUR
z bankomatu

prawda

400 EUR

Konto

```
int _ilość = 600;  
...  
synchronized void  
pobierz(int n){  
    if (_ilość > n){  
        _ilość = _ilość - n;  
    }  
}
```

400

Wyślij 500EUR
do innego konta

fałsz



Przykład wątku

Odchudzające się
zwierzęta

Podsumowanie



Zaawansowane pojęcia

- ▶ Wyjątki
 - ▶ Programy bardziej odporne na błędy
- ▶ Wielowątkowość
 - ▶ Lepsza wydajność
 - ▶ Bardziej zaawansowane programy

**Do zobaczenia
za tydzień**

