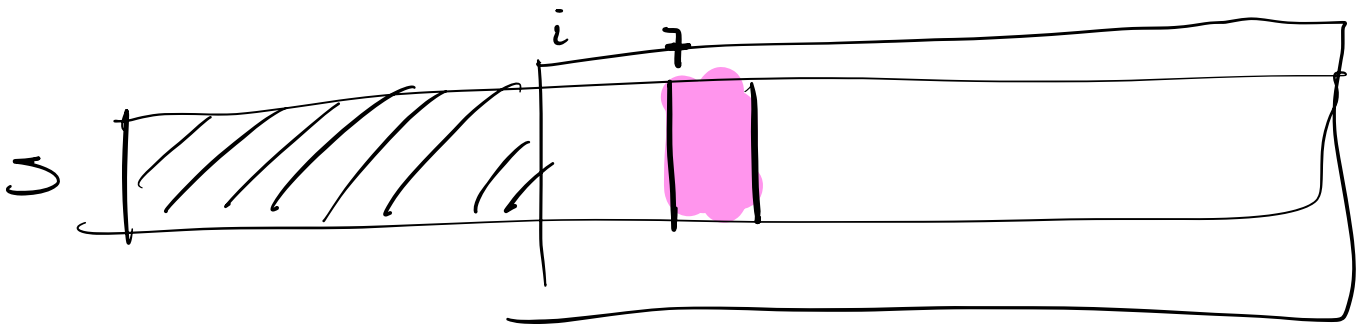


OCCORRENZE (ESERCIZIO)

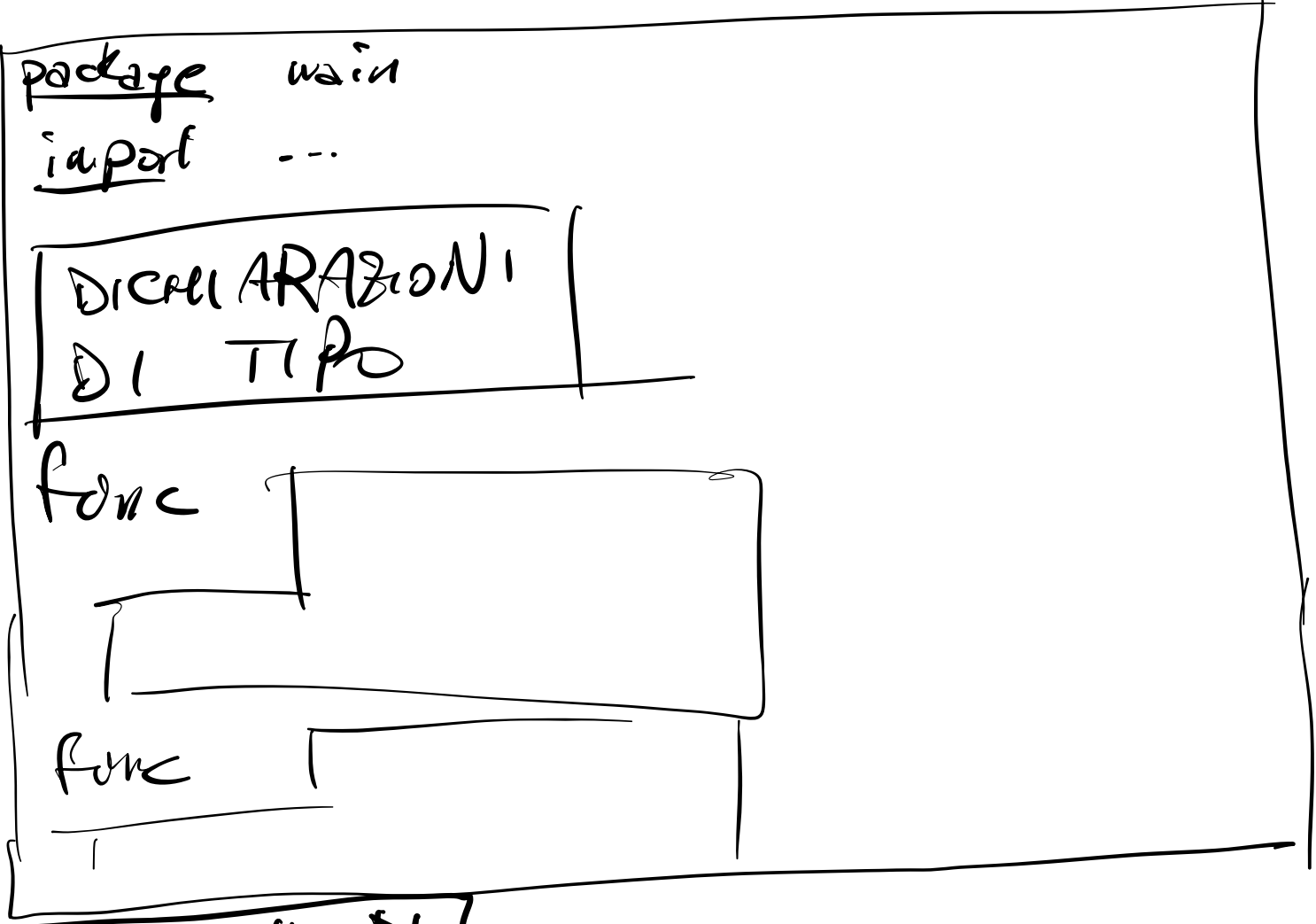
```
func stampaOccorrenze(s, substr string) {  
  var i int  
  for {  
    j := strings.Index(s[i:], substr)  
    if j < 0 {  
      return  
    }  
    fmt.Println(i+j)  
    i = j + 1  
  }  
}
```

↓



TYPE

var x *int
var y ***int



DEFINIZIONI DI TIPO

type

new
tipo

tipo

ESEMPIO

type triplo **int
var x, y triplo
func f (s string, *triplo) triplo
...

↳

ALIAS DI TIPO



<u>type</u>		intero = <u>iat</u>
<u>var</u>	x	intero
<u>var</u>	y	int

$x = y + 5$

<u>type</u>		intero	<u>iat</u>
<u>var</u>	x	intero	
<u>var</u>	y	int	

$y = \text{int}(x)$

$x = \text{intero}(y)$

STRUCT

```
type data struct {  
    g, m, a int
```

```
}
```

type

studente

```
struct {
```

nome, cognome

string

data Di Nascita

data

matricola

string

media

float64

```
}
```

var

x, y

studente

x

nome	cognome	data Di Nascita			matricola	media
		g	m	a		

DOT - NOTATION

x.nome = "Paolo"

x.cognome = "Baldi"

x.dataDiNascita.g = 29

x.dataDiNascita.m = 11

x.dataDiNascita.a = 1968

x.matricola = "359379"

x.media = 18.3

$$y = x$$

ESERCIZIO: scrivete una
funzione che dati due
studenti restituisca
il cognome del più
basso (\Rightarrow media più alta)

```

func piuBravo (s1, s2 studente) string {
  if s1.media > s2.media {
    return s1.cognome
  } else {
    return s2.cognome
  }
}

```

```

func main () {
  var x1, x2 studente
  x1.nome = ---
  ---
  x2.nome = ---
  ---
  fmt.Println (piuBravo (x1, x2))
}

```

ESERCIZIO: Scrivi una funzione
che data uno studente ^{nc}
cambia il nome in "Mario".

MODO 1:

```
func mario (s studente) studente {  
    var t studente  
    t = s  
    t.name = "Mario"  
    return t  
}
```

```
func main () {  
    ...  
    var x studente  
    x.name = "Pippo"  
    x.cognome = "..."  
    ...  
    x = mario(x)  
    ...  
}
```


MODULO 2 :

```
func main (p *studente) {  
    (*p).nome = "Mario"  
}
```

```
func main() {  
    ...  
    var x studente  
    x.nome = "Pippo"  
    x.cognome = "..."  
    ...  
    Mario (&x)  
    ...  
}
```

IN C

$(*p).cognome$

$p \rightarrow cognome$

IN GO

$(*p).cognome$

$p.cognome$

LETTERALI STRUCT

var d data
var s studente

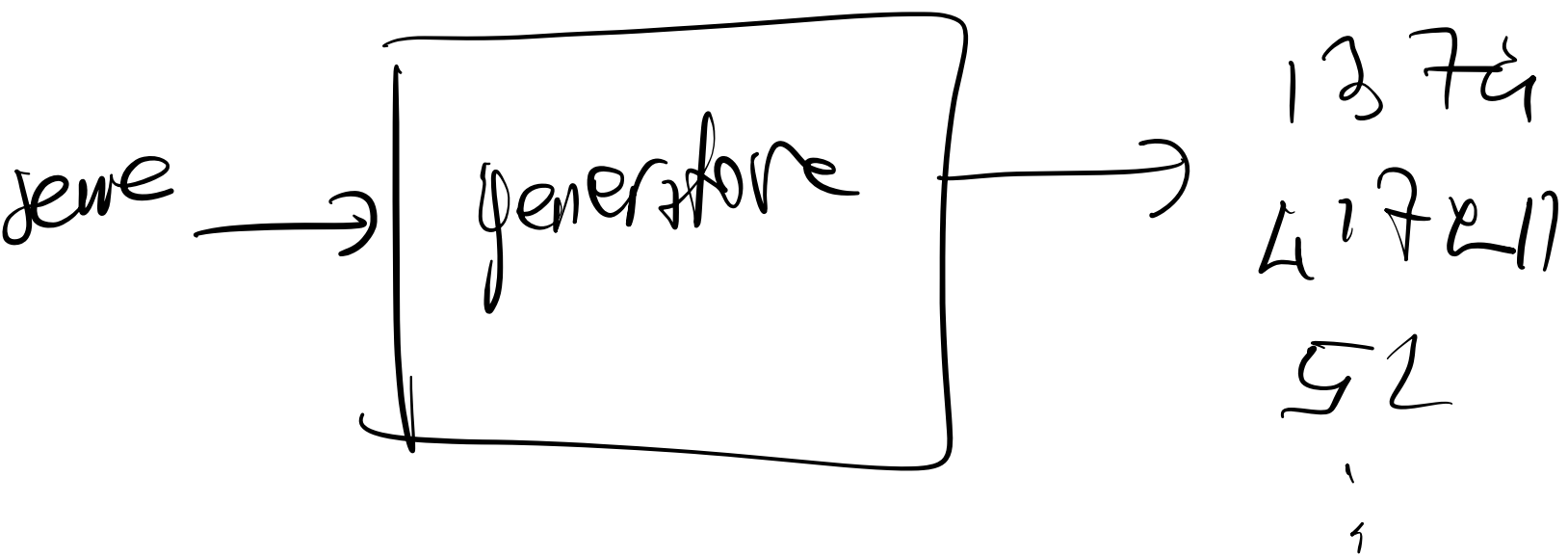
d = data { 29, 11, 1968 }

s = studente { "Paolo",
"Baldi",
data { 29, 11, 1968 },
"352372",
18.6
}

s = studente { nome: "Paolo",
matricola: "374958"
}

NUMERI PSEUDO CASUALI

GENERATORI DI
NUMERI PSEUDO CASUALI
(PRNG)



// Step 0 5 numeri 2 caso
// Step 1 10 da

// File 0 e 10

```
import (
    "fmt"
    "math/rand"
)
func main() {
    for i := 0; i < 5; i++ {
        fmt.Println(rand.Intn(11))
    }
}
```

ESERCIZIO

- Usate la funzione scritta a kune seth use fa per generare e stampare 5 carte da poker