



Forelesning 6

Bugs og Objekter

INF101 - 2023





Hva har disse til felles?





Hva er software bugs?

- Koden gjør ikke det som var tiltenkt.
 - Misforståelser er ofte årsaken til bugs
- Mange bugs er små og uviktige
 - Noen er katastrofale
- Hva kan vi gjøre for å unngå bugs?
- Grace Hopper får æren for «bug».





Høynivå og lavnivå språk

```
public void doTaxes() {  
    Form taxForm = downloadForm();  
    taxForm.fill();  
    taxForm.send();  
}
```

```
section    .text  
    global _start  
_start:  
    mov     edx,len  
    mov     ecx,msg  
    mov     ebx,1  
    mov     eax,4  
    int     0x80  
  
    mov     eax,1  
    int     0x80  
  
section    .data  
msg db 'Hello, world!', 0xa  
len equ $ - msg
```

Hvilket type språk tror du
at du får gjort mest med?





Hvordan skrive god kode?

- Koden er lett å forstå.
 - Gode navn på variabler og typer
 - Bruke hjelpemetoder
- Koden er uten bugs
 - Klart definert hva koden skal gjøre
 - Koden er testet
- Koden er lett å jobbe videre med
 - Legge til ny funksjonalitet uten å ødelegge noe
 - Gjenbruk av kode
- Koden er effektiv
 - Mer om dette i INF102





Hvorfor Java?

- En forskjell mellom java og python
int a = 7;
a = 7
- Hvorfor kan det være bra å vite typen?





Type error in Python

```
def calculate(a, b):  
    print(a + b)
```

```
File "C:/Users/mva021/OneDrive - University of  
Bergen/INF101/variable_type_example.py", line 8, in  
calculate  
    print(a + b)
```

```
TypeError: unsupported operand type(s) for +: 'int'  
and 'str'
```





Type error in Java

```
4 public static void main(String[] args) {  
5     calculate(5, "7");  
6  
7 }  
8  
9  
10 public void calculate(int a, int b) {  
11     System.out.println(a+b);  
12 }
```

The method `calculate(int, int)` in the type `typeCheck` is not applicable for the arguments `(int, String)`

2 quick fixes available:

- Change method '`calculate(int, int)`' to '`calculate(int, String)`'
- Create method '`calculate(int, String)`'

Press 'F2' for focus





Hvilke programmer har bugs?

- Så å si all software har bugs
- I mange prosjekter blir 90% av tiden brukt til debugging
- MS Paint er et enkelt tegneprogram som kommer med Windows
 - Mer enn 100 millioner brukere
 - Første versjon i 1985
 - .Net versjon kom i 2004
- Bug ble rapportert i 2009
- Hva er status 14 år senere?





Vi skal se på kode

- Koden finnes på:
<https://git.app.uib.no/ii/inf101/23v/students/lecture-code>
- Kan vi klare å finne ut hva som er feil med denne koden?
- Hvordan skal vi unngå slike feil?





Hva er Objekter

- Et objekt er en samling med flere variabler som hører sammen.
 - Feltvariabler
- En klasse er koden som beskriver objekter av en bestemt type.
 - F.eks. klassen Person kan brukes til å lage mange Person Objekter
- Konstruktør er metoden som lager nye Objekter
- Objekter har metoder som bruker feltvariablene





Objekt terminologi

- Type
 - hvilken klasse ble brukt til å lage objektet
- Behavior
 - Hvilke metoder kan brukes på Objektet og hva gjør disse metodene?
- Identity
 - Hvilket objekt er det? (Minneadresse)
- State
 - Hvilke data lagrer Objektet akkurat nå





Objektorientert programmering



Abstraction

- Fokuser på det viktigste og gjør det enkelt



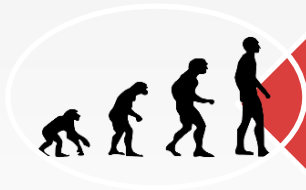
Modularity

- Lag kode som er byggeklosser



Encapsulation

- Skjul detaljer, koden blir lettere å bruke/tryggere



Gjenbruk av kode

- Arv - Gjenbruk av klasser
- Polymorphism – metoder virker på flere typer

SIDE
13





Abstraksjon

- Når man skal lage et program er det ofte begreper og objekter i virkeligheten som skal representeres som objekter i koden.
- Vi representerer ikke alle detaljer, bare det viktigste.





Abstraksjon

- En hver klasse har et visst ansvar, å legge rett funksjonalitet til rett klasse er viktig.
- Når rett funksjonalitet er i rett klasse blir systemet lett å bruke og lett å forstå
- Ikke for mye detaljer, ikke for lite funksjonalitet.





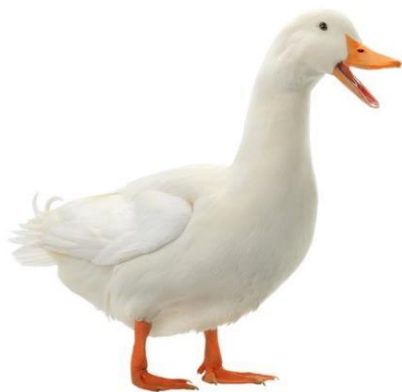
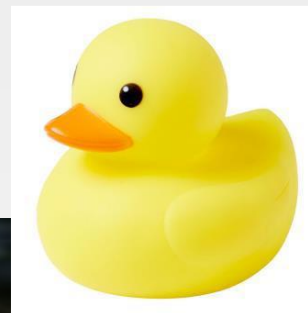
Abstraksjon

- Abstraksjon handler også om å finne ut hva som er felles for flere klasser
- Felles kode bør trekkes ut og gjenbrukes.





Abstraksjon





Sitat – duck-typing

James Whitcomb Riley (1849–1916):

“When I see a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.”

Duck-typing betyr at vi trenger ikke alltid beskrive virkelige objekter nøyaktig, bare de egenskapene som er nødvendige.





Objekt klassen

- Alle klasser er av typen Objekt
 - Vi lærer mer om dette når vi kommer til arv
- Noen viktige metoder finnes i alle klasser
 - toString()
 - equals()
- Forrige uke lærte dere forskjellen på equals() og ==





La oss implementere Bibliotek

- Biblioteket trenger et system for å holde styr på alle bøkene sine.
- Kan vi hjelpe dem?
- Lage en klasse Book
- Lage et system for å holde styr på mange bøker





```
public class Thanks{
    private String text;

    public Thanks(String text){
        this.text = text;
    }

    public String toString(){
        return this.text;
    }

    public static void main(String[] args){
        Thanks end = new Thanks("Takk for forelesning!");
        System.out.println(end);
    }
}
```

