



# const:t on monet

5.12.2018

# const - vakiot

- Vakiot “muuttujia”, joita ei voi muuttaa
- Käännösaikaiset vakiot: `int taulukko[MAX];`
- Normaalisti yhden käännösyksikön sisäisiä
- Yleensä otsikkotiedostoissa



# const - perustyytit

```
int const MAX_MJONON_KOKO = 30000;  
double const PI = 3.14159265;  
int const RAJA = annaRaja();  
char mjonon[MAX_MJONON_KOKO];
```



# Vakio-oliot

- Olioista vakioita samoin kuin muuttujista:  
    Paivays **const** joulu(24,12,1999);
- Mitä tarkoittaa “vakio-olio”?
- Sijoitus?



# Vakio-oliot

- Vakio-olio: olion **tilaa** ei saa muuttaa
- Jäsenfunktioiden jako: “muuttavat” jäsenfunktiot & vakiojäsenfunktiot
- Kaksi rajapintaa
- Vakiojäsenfunktion tunnus **const** parametrilistan jälkeen



# Päiväysluokka

## const:ineen

```
class Paivays {
public:
    Paivays(unsigned int p, unsigned int k,
unsigned int v);
    ~Paivays();

    void asetaPaiva(unsigned int paiva);
    void asetaKk(unsigned int kuukausi);
    void asetaVuosi(unsigned int vuosi);

    unsigned int annaPaiva() const;
    unsigned int annaKk() const;
    unsigned int annaVuosi() const;

    void etene(int n);
    int paljonkoEdella(Paivays const& p)
const;
```

```
private:
    unsigned int
paiva_;
    unsigned int
kuukausi_;
    unsigned int
vuosi_;
```

3.10.2018

# Päiväysluokka const:ineen

```
void Paivays::asetapaiva(unsigned int paiva)
{
    paiva_ = paiva;
}
```

```
unsigned int Paivays::annaPaiva() const
{
    return paiva_;
}
```

# Kääntäjän const- varmistukset

- Vakiojäsenfunktion koodissa rajoituksia
  - Jäsenmuuttujat
  - Jäsenfunktiot
  - **this**





# Kääntäjän const- varmistukset

- Olion tila = jäsenmuuttajat?
- Olion tila  $\subset$  jäsenmuuttajat?
- Olion tila  $\supset$  jäsenmuuttajat?

# Vakioviitteet ja -osoittimet

- Vakio-oliot harvinaisia, muutos osa olioiden luonnetta
- Vakioviite: `Paivays const& p;`
- Vakio-osoitin: `char const* miono;`
- (Osoitinvakio: `char* const miono;`)

# Vakioviitteet ja -osoittimet

- Osoitettu/viitattu olio/muuttuja **käyttäytyy** kuin vakio
- Mahdollisuus rajata ei-vakio-olion rajapinta vakioksi

# Vakioviitteet ja -osoittimet

- Dokumentointikeino
- Suojausmekanismi

```
void muutanKuitenkin(Paivays const& pvm) {  
    pvm.asetapaiva(1); // KÄÄNNÖSVIRHE:  
                        // asetaPaiva ei  
vakiojäsenfunktio
```

# Muunnokset vakioksi muttei takaisin

- C++:ssa on implisiittinen tyyppimuunnos: normaaliosoitin → vakio-osoitin
- **Ei** ole implisiittistä tyyppimuunnosta: vakio-osoitin → normaaliosoitin

```
char* miono = "Käytä string-luokkaa char*:n sijaan";  
char const* vakiomiono = miono; // Ok: ei-vakio → vakio  
char* miono2 = vakiomiono; // KÄÄNNÖSVIRHE: vakio → ei-vakio
```

# const-sanan tärkeydestä

- Lisää kääntäjän varmistusmahdollisuuksia
- Dokumentointikeino
- Vakio-olioita vain vakio-osoittimen/-viitteen päähän
- Yksi **const** jää pois → muidenkin käyttö vaikeutuu

# const-virhketju

```
#include "paivays.hh"
```

```
bool onkoKarkauspaivaa(Paivays* pvm_p);
```

```
bool saakoHameenKuukaudessa(Paivays const& nyt)
```

```
{ // Karkauspäivä on kuukauden sisällä, jos on helmikuu ja vuosi  
on karkausvuosi
```

```
    if (nyt.annaKk() != 2) {
```

```
        return false; // Ei helmikuu
```

```
    }
```

```
    else {
```

```
        return onkoKarkauspaivaa(&nyt); // KÄÄNNÖSVIRHE
```

```
    }
```

