

Cvičenie 12

Obsah

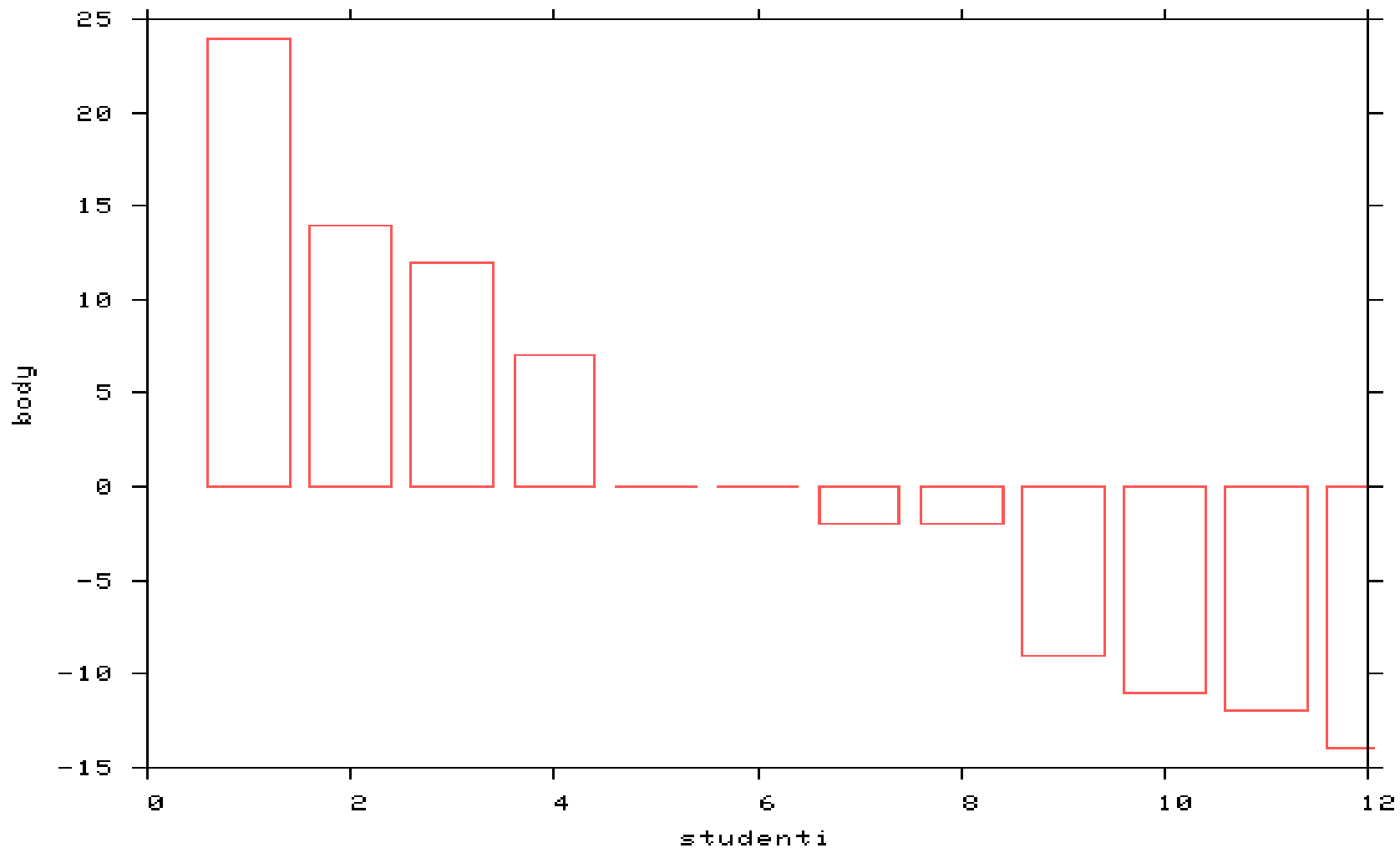
- organizačné záležitosti
- šablony

Organizačné záležitosti

- úloha 10 (textové ovládacie prvky) – deadline o týždeň
- oprava úloh a udelenie zápočtov
 - mail
- program na posledné cvičenie?

Organizačné záležitosti

- aktuálna štatistika



Šablony

- poznáte (STL)
- instancionalizace
- ďalšia rovina abstrakcie (šablona, trieda, objekt)
- nový koncept (neimperatívne)
- široké možnosti

Šablony

- definované v hlavičkovom súbore
- *this*
- základná syntax:

template<typename T> class [nazov] {};

template<int size> class A {};

template<int p1, typename p2> class A {};

template<typename p2 = double> class A {};

template<typename T, template <typename> class B> class A

- demo12_1.cc

Šablony funkcí

- umožňuje psát bezpečné generické funkce
- kompilátor sám zistuje a vyhodnocuje parametre šablony
 - demo12_2.cc
- inštancie sa tvoria len v prípade potreby

Šablony funkcí

- určujú isté štandardné správanie, pomocou špecializácií môžem vytvoriť výnimky
- špecializácia
 - vytvorenie špeciálnej šablony pre isté hodnoty
 - môže byť:
 - úplná (explicitná)
 - pre konk. hodnoty parametrov
 - čiastočná
 - pre nejakú podmnožinu parametrov (fixujeme len nejaké parametre, konk. typ)

Šablony funkcí

- postup kompilátora při volání funkce
 - klasická funkce splňující prototyp
 - úplná specializace šablony
 - šablona pro daný typ

- demo12_3.cc

Šablony metód

- podobne ako funkcie

- demo12_4.cc

Šablony tried

- umožňujú typovú nezávislosť a generické typy
- metódy automaticky majú šablonu triedy
- ukážka čiastočnej a úplnej špecializácie šablon tried
 - demo12_5.cc
- inak rovnaké pravidlá ako pri šablonách funkcií
 - počet parametrov, default, friend, manipulácia...

V čom spočíva sila šablon?

- určenie niečoho v dobe prekladu v závislosti od platformy
 - typ/niečo, čo je zbytočné počítat' pri každom behu
 - `demo12_6.cc`
- metaprogramovanie
 - programy vykonávané pri kompilácii
 - možnosť využiť preprocesor
 - pomerne náročné, ale otvára nové možnosti
 - podobné logickému programovaniu
 - výsledkom je konštanta využívaná pri behu

Metaprogramovanie

- typický princíp
 - rekurzia + zarážka vytvorená špecializáciou
 - demo12_7.cc
- umožňuje nám zvýšiť efektivitu (prečo?), ale aj vytvoriť statické náhrady za konštrukcie jazyka
 - if, switch, cyklus...
 - demo12_7.cc
 - demo12_8.cc

Zdroje týchto materiálov a ďalšie info

- prednáška z PB161 (Mgr. Šimon Tóth)

<http://www.simontoth.cz/cs/prednaska-sablony>

– k dispozícii video, slidy, demá...

Program na posledné cvičenie?

Dotazy?