Zavod za elektroniku, mikroelektroniku
računalne i inteligentne sustave

23. ožujka 2021.

Oblikovni obrasci u programiranju

1. laboratorijska vježba

1. Vaše rješenje u prvoj vježbi trebalo je omogućiti ispravno izvršavanje sljedeće ispitne funkcije:

```
void testAnimals(void){
  struct Animal* p1=createDog("Hamlet");
  struct Animal* p2=createCat("Ofelija");
  struct Animal* p3=createDog("Polonije");

  animalPrintGreeting(p1);
  animalPrintGreeting(p2);
  animalPrintGreeting(p3);

  animalPrintMenu(p1);
  animalPrintMenu(p2);
  animalPrintMenu(p3);

  free(p1); free(p2); free(p3);
}
```

   Predložiti izvorni kod u programskom jeziku C koji bi omogućio proširivanje ove ispitne funkcije prema sljedećim zahtjevima:

   (a) dodati novu virtualnu metodu svim izvedenim razredima razreda `Animal`. Nova virtualna metoda nam otkriva čime se životinje vole igrati; mačke s "klupkom", a psi s "lopticom";

   (b) dodati novi izvedeni razred `Cow` koji modelira životinju koja pozdravlja s "muuu", voli jesti "novine" i igrati "nogomet".

2. Konstruirajte minimalni kod u C++ čije bi prevođenje dovelo do ovakvog fragmenta strojnog koda. Strojni kod se nalazi u tijelu članske funkcije razreda `Client`, a memorijske lokacije na pomacima -8 i -16 od rbp-a sadrže argumente funkcijskog poziva.

```
mov     rdi, QWORD PTR -8[rbp]
call    _ZN4BaseC2Ev
lea     rdx, _ZTV6Client
mov     rax, QWORD PTR -8[rbp]
mov     QWORD PTR [rax], rdx
mov     rax, QWORD PTR -16[rbp]
mov     rax, QWORD PTR [rax]
mov     rdx, QWORD PTR [rax]
mov     rdi, QWORD PTR -16[rbp]
call    rdx
```

3. Napravljena je modifikacija 2. zadatka laboratorijske vježbe, dodavanjem razreda `LinearSquared` te novom metodom `main`.

   (a) Za prikazani programski kod prikažite stanje u memoriji računala u trenutku prije izvođenja retka koji ispisuje tekst "Gotovo". Pod stanjem u memoriji podrazumijevamo detaljan prikaz svih varijabli, objekata, tablica, pokazivača i slično. Posebno naznačite što se od toga nalazi na stogu, a što na gomili. Pomoć: tablice virtualnih funkcija te izvršni kod metoda ne nalaze se ni na stogu ni na gomili, nego u zasebnim memorijskim segmentima.

   (b) Za svaki od poziva metode `value_at` u metodi main napišite koje je stvarno odredište (i objasnite). Odredite što će biti ispisano prije no što se ispiše gotovo. Objasnite kako se došlo do ispisanih brojeva.

```
class Unary_Function {
  private:
    int lower_bound;
    int upper_bound;
  public:
    Unary_Function(int lb, int ub) : lower_bound(lb), upper_bound(ub) {};
    virtual double value_at(double x) = 0;
    virtual double negative_value_at(double x) {
      return -value_at(x);
    }
    void tabulate() {
      for(int x = lower_bound; x <= upper_bound; x++) {
        printf("f(%d)=%lf\n", x, value_at(x));
      }
    };
    static bool same_functions_for_ints(Unary_Function *f1, Unary_Function *f2, double tolerance) {
      if(f1->lower_bound != f2->lower_bound) return false;
      if(f1->upper_bound != f2->upper_bound) return false;
      for(int x = f1->lower_bound; x <= f1->upper_bound; x++) {
        double delta = f1->value_at(x) - f2->value_at(x);
        if(delta < 0) delta = -delta;
        if(delta > tolerance) return false;
      }
      return true;
    };
};
class Square : public Unary_Function {
  public:
    Square(int lb, int ub) : Unary_Function(lb, ub) {};
    virtual double value_at(double x) {
      return x*x;
    };
};
class Linear : public Unary_Function {
  private:
    double a;
    double b;
  public:
    Linear(int lb, int ub, double a_coef, double b_coef) : Unary_Function(lb, ub), a(a_coef), b(b_coef) {};
    virtual double value_at(double x) {
      return a*x + b;
    };
};
class LinearSquared : public Linear {
  public:
    LinearSquared(int lb, int ub, double a_coef, double b_coef) : Linear(lb, ub, a_coef, b_coef) {};
    virtual double value_at(double x) {
      double d = Linear::value_at(x);
      return d*d;
    };
};
int main() {
  LinearSquared ls[2] = {LinearSquared(-2,+2,1,2),LinearSquared(-2,+2,3,4)};
  Linear* pls = new LinearSquared(-2,+2,5,6);
  cout << ls[1].value_at(0) << endl;
  Linear* l1 = &ls[1];
  cout << l1->value_at(0) << endl;
  Unary_Function *l2 = pls;
  cout << l2->value_at(0) << endl;
  l1->tabulate();
  cout << "Gotovo" << endl;
}
```

Rješenje zadatka 1:

```
// a)
char const* dogGame(){
  return "lopticom";
}
char const* catGame(){
  return "klupkom";
}

PTRFUN vtblDog[3]={dogGreet,dogMenu,dogGame};
PTRFUN vtblCat[3]={catGreet,catMenu,catGame};

void animalPrintGame(struct Animal* p){
  printf("␣␣%s␣voli␣igrati␣%s\n", p->name, p->vtable[2]());
}
// ================================================
// b)
char const* cowGreet(){
  return "muuu!";
}
char const* cowMenu(){
  return "novine";
}
char const* cowGame(){
  return "nogomet";
}

PTRFUN vtblCow[3]={cowGreet,cowMenu,cowGame};

struct Animal* constructCow(struct Animal* panimal, char const* realName){
  panimal->vtable=&vtblCow[0];
  panimal->name=realName;
  return panimal;
}

struct Animal* createCow(char const* realName){
  struct Animal* panimal=(struct Animal*)malloc(sizeof(struct Animal));
  panimal = constructCow(panimal, realName);
  return panimal;
}

// ostalo
typedef char const* (*PTRFUN)();

struct Animal{
  char const* name;
  PTRFUN* vtable;
};

void animalPrintGreeting(struct Animal* p){
  printf("␣␣%s␣pozdravlja:␣%s\n", p->name, p->vtable[0]());
}
void animalPrintMenu(struct Animal* p){
  printf("␣␣%s␣voli␣%s\n", p->name, p->vtable[1]());
}
```

Rješenje zadatka 2:

```
class Base{
  int m_;
public:
  Base(){
    m_=0;
  }
  virtual void m(){}
};

class Worker{
public:
  virtual int m(){return 42;}
};

class Client: public Base{
public:
  Client(Worker* pw){
    pw->m();
  }
};

int main(){
  Worker w;
  Client c(&w);
}
```

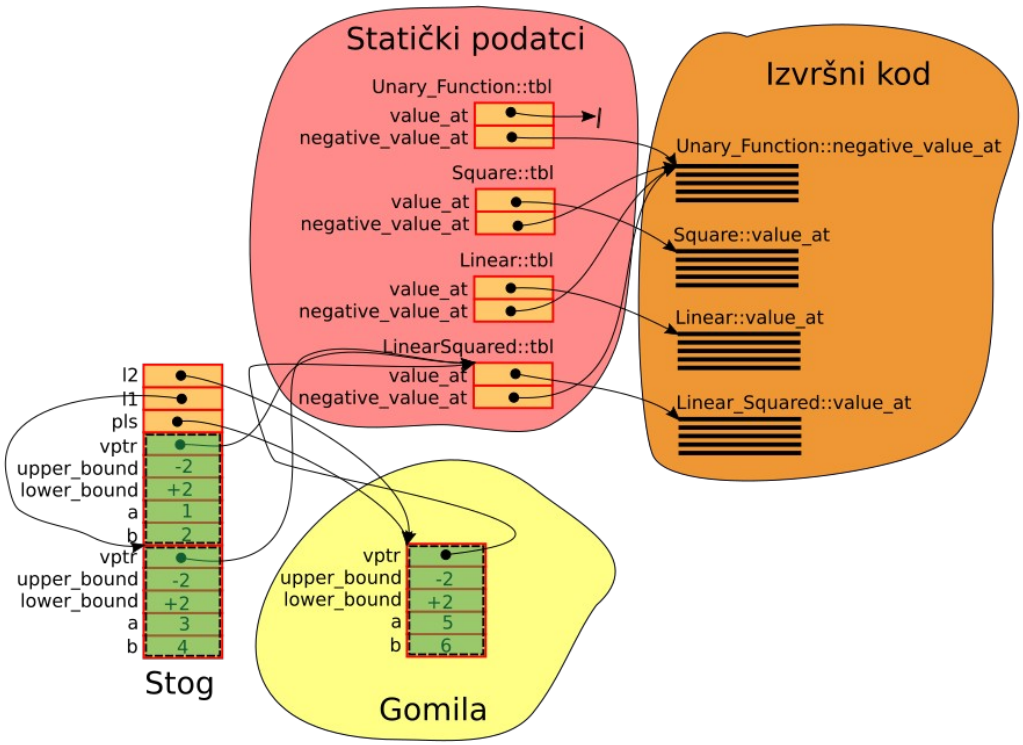
Rješenje za sve bodove treba:

- skužiti da tijelo metode ima virtualni poziv `p->m()` (4)
- skužiti da su argumenti metode i) this i ii) pokazivač na objekt p (2)
- skužiti da se radi o tijelu konstruktora razreda Client izvedenog iz Base (2)
- imati barem jednu virtualnu metodu u razredu Client ili Base (1)
- imati definiciju razreda Worker s virtualnom metodom m() bez argumenata (1)

Rješenje zadatka 3:

Zadatak 3.

a) Stanje u memoriji



Bodovanje: 5 bodova (a) dio

- polje je na stogu: 1 bod
- objekt alociran s new je na gomili: 1 bod
- napisane i ispravno popunjene tablice svih razreda: 1 bod
- kod objekata prisutan pointer na tablicu virtualnih funkcija i ispravno postavljen: 1 bod
- korektan sadržaj svih deklariranih pokazivača te članskih varijabli objekata: 1 bod

b) Što će se biti ispisano:

```
16
16
36
4
1
16
49
100
```