

Σ 30 pp

1 a i) 2

ii) public;

iii) Der Zugriffstyp verhindert das Verändern der Variable ✓

b i) 4

ii) };

iii) Ein Semikolon hat gefehlt ✓

c i) 3

ii) Test () { \_i = 0; }

iii) Die Main-Methode übergibt keine Argumente. ✓

d i) 10

ii) p → say.Hello ();

iii) Hier wird der Pointer-Operator benötigt. ✓

12

2 a i) Es wird nur der Zeiger auf die Variable gelöscht, aber nicht der Speicher frei gegeben.

ii) 6 ~ BadClass { delete p; } ✓

b i) Es ist zwar eine Schnittstelle vorhanden, die Variable ist aber vom Zugriffstyp public.

ii) 5 private ✓

6 int value;

c i) str wird auf str gesetzt, was nicht möglich ist. str ist vom Typ const char.

Die private Variable bleibt unverändert.

ii) 5 void setName (const char\* str) {

6 this → str = str; ✓

7 }

9

3) class Vektor2d {

private:

double x;

double y;

public:

Vektor2d ();

Vektor2d (double x, double y);

~ Vektor2d ();

double normsq (); ✓

};

Vektor2d::Vektor2d () {

x = 0;

y = 0; ✓

}

Vektor2d::Vektor2d (double \_x, double \_y) {

this->x = \_x;

this->y = \_y; } ✓

Vektor2d::~~Vektor2d () {} ✓

double Vektor2d::normsq () {

return x\*x + y\*y; } ✓

int main () { return 0; } ?

Schnittstelle = i, iii, iv, vi ✓

g