

Arbeitsblatt 5, infT

Copy Construction und Zuweisung

Gegeben:

```
class Dog {
    string nm;
public:
    Dog(const string& nm) : nm(nm) { cout << "Creating Dog: " << *this << endl; }
    // Create a Dog from a Dog pointer:
    Dog(const Dog *dp, const string& msg) : nm(dp->nm + msg) {
        cout << "Copied dog " << *this << " from " << *dp << endl;
    }
    ~Dog() { cout << "Deleting Dog: " << *this << endl; }
    void rename(const string& nm) {
        this->nm = nm;
        cout << "Dog renamed to: " << *this << endl;
    }
    friend ostream& operator<<(ostream& os, const Dog& d) {
        return os << "[" << d.nm << "]";
    }
};
```

Implementieren Sie eine Klasse DogHouse gemäss:

```
class DogHouse {
    Dog* p; // Zeiger auf Bewohner
    string houseName; // Name der Hundehuette
public:
    // Erzeuge Huette mit Zeiger auf Bewohner und mit Huettentname
    DogHouse(Dog *, const string&);
    // Erzeuge neue Huette mit neuem Bewohner.
    DogHouse(const DogHouse&);
    // Nehme ein Zuweisung vor.
    DogHouse& operator=(const DogHouse& house);
    void renameHouse(const string& nm);
    Dog *getDog() const;
    ~DogHouse();
    friend ostream& operator<<(ostream& os, const DogHouse& dh);
};
```

Hinweise:

- `DogHouse(const DogHouse&)` werde so codiert, dass der Name der Hütte und der Name des Hundes vom Original übernommen werden und die Namen mit String-Concatenation je eine entsprechende Ergänzung erfahren. Für den Hundename z.B in der Art:
`p(new Dog(house.p, " copy-constructed"))`
- `DogHouse& operator=(const DogHouse& house)` werde so codiert, dass mit der Zuweisung ein neuer Bewohner erstellt werde. Der Name der Hütte und der Name des Hundes sollen vom Original übernommen werden und mit String-Concatenation je eine entsprechende Ergänzung erfahren. Für den Hundename z.B in der Art: `p(new Dog(house.p, " assigned"))`
- Es kann nützlich sein, die Verhältnisse mit Objektdiagrammen in Schritten aufzuzeichnen.

Testen Sie Ihren Code mit folgendem main:

```
int main() {
    cout << "Construction of Fido and FidoHuetten" << endl;
    DogHouse fidoHouse(new Dog("Fido"), "FidoHuetten");
    cout << fidoHouse << endl;

    cout << "Copy Construction of Lassie and LassieHuetten" << endl;
    DogHouse fidoHouse2 = fidoHouse;
    cout << fidoHouse2 << endl;
    fidoHouse2.getDog()->rename("Lassie");
    fidoHouse2.renameHouse("LassieHuetten");
    cout << fidoHouse2 << endl;

    cout << "Assignment and renaming to Max and MaxHuetten" << endl;
    fidoHouse = fidoHouse2;
    cout << fidoHouse << endl;
    fidoHouse.getDog()->rename("Max");
    fidoHouse.renameHouse("MaxHouse");

    return 0;
}
```

Erwartete Ausgabe:

```
Construction of Fido and FidoHuetten:
Creating Dog: [Fido]
[FidoHuetten] contains [Fido]

Copy Construction of Lassie and LassieHuetten:
Copied dog [Fido copy-constructed] from [Fido]
[FidoHuetten copy-constructed] contains [Fido copy-constructed]
Dog renamed to: [Lassie]
DogHouse renamed to: [LassieHuetten] contains [Lassie]
[LassieHuetten] contains [Lassie]

Assignment and renaming to Max and MaxHuetten:
Copied dog [Lassie assigned] from [Lassie]
[LassieHuetten assigned] contains [Lassie assigned]
Dog renamed to: [Max]
DogHouse renamed to: [MaxHouse] contains [Max]
Deleting Dog: [Lassie]
Deleting Dog: [Max]
```