

Aufgabe 1) Operatoren

Implementieren Sie bitte die folgenden Operatoren für die Klasse `bitVector` aus Übungszettel 4, Aufgabe 1:

- `+=` : soll den den Bit-Vektor auf der rechten Seite der Zuweisung um den Ausdruck auf der linken Seite der Zuweisung erhöhen (Benutzung der Addition aus Übungszettel 2, Aufgabe 2)
- `++` : soll den angegebenen Bit-Vektor um 1 erhöhen
- `<<` soll analog zu `printBitVector` den Bit-Vektor in Form von ASCII-Zeichen auf einem **ostream** ausgeben
- `>>` soll es erlauben, eine Eingabe aus **istream** der Form „011101001011“ in einen Bitvektor hineinzuschreiben. Achten Sie hier bitte auf die Länge des Strings im Vergleich zur Länge des Bitvektors (da dies kein Konstruktor ist, haben Sie die Wahl: entweder abschneiden oder Länge anpassen)
- Implementieren Sie den Copy-Konstruktor und die Wertzuweisung für die `bitVector`-Klasse
- Implementieren Sie die `+` Operation für die Klasse `bitVector`
- Was müssen Sie tun, um Ausdrücke der Form `bv1 = bv2 + "0001110101"` zu ermöglichen? Implementieren und testen Sie ihr Konzept. Achtung: Probieren Sie es vorher aus, vielleicht kann Ihre Implementierung dies schon (es gibt mehrere Wege, dies zu erreichen).

Achten Sie bitte darauf, wann Sie Referenzen benutzen und wann nicht. Testen Sie die neuen Operationen auch mit komplexen Beispielen, bspw.:

```
int main() {
    bitVector bv1 = "1111001101";
    bitVector bv2 = bv1;
    bitVector bv3;
    bitVector bv4;
    cin >> bv3;
    cout << ++bv1 + bv3 + bv2++;
    cout << bv1 << " " << bv2 << " " << bv3 << endl;
    bv4 << bv1 + "110101010";
    cout << bv4 << endl;
}
```