4.25 Контейнер vector<bool>. Отличие от обычного vector. Класс BoolReference, его реализация. Реализация метода [] в vector<bool>.

Он отличается от обычного вектора тем, что хранит не просто массив буллей, а пакует его в **пачки по 8 логических значений и представляет их как один байт.** (То есть на одно значние приходится 1 бит)

В vector<bool> интересно работает присваивание.

```
templtate <typename U>
void f(const U&) = delete;

int main() {
    vector<bool> vb(10, false);
    vb[5] = true;
    f(vb[5]);
}
```

В данном случае компилятор начнет ныть, что нельзя вызывать f от типа, который удален. Но так мы заставим компилятор спалить какой у него тип для vb[5].
Мы увидим, что **U = std::_Bit_reference**. Как же это работает?

```
templtate <>
class Vector<bool> {
    int8_t* arr;
    size_t sz;
    size_t cap;

    struct BitReference {
        int8_t* cell;
        uint8_t num; // pos in this cell

        BitReference& operator=(bool b) {
            if (b) {
                *cell |= (1u << num);
            } else {
                *cell &= ~(1u << num);
            }
            return *this;
        }

        operator bool() const {
            return *cell & (1u << num);
        }
    }

public:
    BitReference operator[](size_t i) {
        return BitReference{arr + i / 8, i % 8};
    }
}
```

Структура BitReference такаия хитрая, что она позволяет, присваиваяя экземпляру себя, менять исходный вектор.