

GildedRose - Adam Trawiński

Kod przed refaktoryzacją

```
for (let i = 0; i < this.items.length; i++) {
    if (this.items[i].name != 'Aged Brie' && this.items[i].name != 'Backstage passes
to a TAFKAL80ETC concert') {
        if (this.items[i].quality > 0) {
            if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
                this.items[i].quality = this.items[i].quality - 1
                if (this.items[i].name.lastIndexOf('Conjured ', 0) === 0 && this.
items[i].quality > 0) {
                    this.items[i].quality = this.items[i].quality - 1;
                }
            }
        }
    } else {
        if (this.items[i].quality < 50) {
            this.items[i].quality = this.items[i].quality + 1
            if (this.items[i].name == 'Backstage passes to a TAFKAL80ETC concert') {
                if (this.items[i].sellIn < 11) {
                    if (this.items[i].quality < 50) {
                        this.items[i].quality = this.items[i].quality + 1
                    }
                }
                if (this.items[i].sellIn < 6) {
                    if (this.items[i].quality < 50) {
                        this.items[i].quality = this.items[i].quality + 1
                    }
                }
            }
        }
    }
}
if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
    this.items[i].sellIn = this.items[i].sellIn - 1;
}
if (this.items[i].sellIn < 0) {
    if (this.items[i].name != 'Aged Brie') {
        if (this.items[i].name != 'Backstage passes to a TAFKAL80ETC concert') {
            if (this.items[i].quality > 0) {
                if (this.items[i].name != 'Sulfuras, Hand of Ragnaros') {
                    this.items[i].quality = this.items[i].quality - 1
                    if (this.items[i].name.lastIndexOf('Conjured ', 0) === 0 &&
this.items[i].quality > 0) {
                        this.items[i].quality = this.items[i].quality - 1;
                    }
                }
            }
        }
    }
}
```

```

        } else {
            this.items[i].quality = this.items[i].quality - this.items[i].quality
        }
    } else {
        if (this.items[i].quality < 50) {
            this.items[i].quality = this.items[i].quality + 1
        }
    }
}

return this.items;

```

Kroki refaktoryzacji

Krok 1: Zmienić typ pętli i wykorzystanie elementów pętli

```

for (const item of this.items) {
    if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
        if (item.quality > 0) {
            if (item.name !== 'Sulfuras, Hand of Ragnaros') {
                item.quality = item.quality - 1
                if (item.name.lastIndexOf('Conjured ', 0) === 0 && item.quality > 0) {
                    item.quality = item.quality - 1;
                }
            }
        }
    } else {
        if (item.quality < 50) {
            item.quality = item.quality + 1
            if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
                if (item.sellIn < 11) {
                    if (item.quality < 50) {
                        item.quality = item.quality + 1
                    }
                }
                if (item.sellIn < 6) {
                    if (item.quality < 50) {
                        item.quality = item.quality + 1
                    }
                }
            }
        }
    }
}

if (item.name !== 'Sulfuras, Hand of Ragnaros') {
    item.sellIn = item.sellIn - 1;
}

if (item.sellIn < 0) {

```

```

        if (item.name != 'Aged Brie') {
            if (item.name != 'Backstage passes to a TAFKAL80ETC concert') {
                if (item.quality > 0) {
                    if (item.name != 'Sulfuras, Hand of Ragnaros') {
                        item.quality = item.quality - 1
                        if (item.name.lastIndexOf('Conjured ', 0) === 0 && item
.quality > 0) {
                            item.quality = item.quality - 1;
                        }
                    }
                }
            } else {
                item.quality = item.quality - item.quality
            }
        } else {
            if (item.quality < 50) {
                item.quality = item.quality + 1
            }
        }
    }
}

return this.items;

```

Krok 2: Deklaracja funkcji zmniejszającej parametr "Quality"

```

decreaseQuality(item: Item) {
    if (item.quality > 0) {
        if (item.name != 'Sulfuras, Hand of Ragnaros') {
            item.quality = item.quality - 1
            if (item.name.lastIndexOf('Conjured ', 0) === 0 && item.quality > 0) {
                item.quality = item.quality - 1;
            }
        }
    }
}

```

Krok 3: Podstawienie funkcji z kroku 2

```

for (const item of this.items) {
    if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
        this.decreaseQuality(item);
    } else {
        if (item.quality < 50) {
            item.quality = item.quality + 1
            if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
                if (item.sellIn < 11) {
                    if (item.quality < 50) {
                        item.quality = item.quality + 1
                    }
                }
                if (item.sellIn < 6) {
                    if (item.quality < 50) {
                        item.quality = item.quality + 1
                    }
                }
            }
        }
    }
}

if (item.name !== 'Sulfuras, Hand of Ragnaros') {
    item.sellIn = item.sellIn - 1;
}

if (item.sellIn < 0) {
    if (item.name !== 'Aged Brie') {
        if (item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
            this.decreaseQuality(item);
        } else {
            item.quality = item.quality - item.quality
        }
    } else {
        if (item.quality < 50) {
            item.quality = item.quality + 1
        }
    }
}

}

return this.items;

```

Krok 4: Deklaracja funkcji zwiększającej parametr "Quality" dla przedmiotów typu "Backstage"

```

backstageIncreaseQuality(item: Item) {
  if (item.sellIn < 11) {
    if (item.quality < 50) {
      item.quality = item.quality + 1
    }
  }
  if (item.sellIn < 6) {
    if (item.quality < 50) {
      item.quality = item.quality + 1
    }
  }
}

```

Krok 5: Podstawienie funkcji z kroku 4

```

for (const item of this.items) {
  if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
    this.decreaseQuality(item);
  } else {
    if (item.quality < 50) {
      item.quality = item.quality + 1
      if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
        this.backstageIncreaseQuality(item);
      }
    }
  }
  if (item.name !== 'Sulfuras, Hand of Ragnaros') {
    item.sellIn = item.sellIn - 1;
  }
  if (item.sellIn < 0) {
    if (item.name !== 'Aged Brie') {
      if (item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
        this.decreaseQuality(item);
      } else {
        item.quality = item.quality - item.quality
      }
    } else {
      if (item.quality < 50) {
        item.quality = item.quality + 1
      }
    }
  }
}

return this.items;

```

Krok 6: Deklaracja funkcji zmniejszającej parametr "SellIn"

```

decreaseSellIn(item: Item) {
  if (item.name != 'Sulfuras, Hand of Ragnaros') {
    item.sellIn = item.sellIn - 1;
  }
}

```

Krok 7: Podstawienie funkcji z kroku 6

```

for (const item of this.items) {
  if (item.name != 'Aged Brie' && item.name != 'Backstage passes to a TAFKAL80ETC concert') {
    this.decreaseQuality(item);
  } else {
    if (item.quality < 50) {
      item.quality = item.quality + 1
      if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        this.backstageIncreaseQuality(item);
      }
    }
  }
  this.decreaseSellIn(item);
  if (item.sellIn < 0) {
    if (item.name != 'Aged Brie') {
      if (item.name != 'Backstage passes to a TAFKAL80ETC concert') {
        this.decreaseQuality(item);
      } else {
        item.quality = item.quality - item.quality
      }
    } else {
      if (item.quality < 50) {
        item.quality = item.quality + 1
      }
    }
  }
}

return this.items;

```

Krok 8: Przeniesienie funkcji zwiększającej parametr "Quality" dla typu "Backstage"

```

for (const item of this.items) {
  if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
    this.decreaseQuality(item);
  } else {
    if (item.quality < 50) {
      item.quality = item.quality + 1
    }
    if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
      this.backstageIncreaseQuality(item);
    }
  }
  this.decreaseSellIn(item);
  if (item.sellIn < 0) {
    if (item.name !== 'Aged Brie') {
      if (item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
        this.decreaseQuality(item);
      } else {
        item.quality = item.quality - item.quality
      }
    } else {
      if (item.quality < 50) {
        item.quality = item.quality + 1
      }
    }
  }
}

return this.items;

```

Krok 9: Deklaracja funkcji zwiększającej parametr "Quality"

```

increaseQuality(item: Item) {
  if (item.quality < 50) {
    item.quality = item.quality + 1
  }
}

```

Krok 10: Podstawienie funkcji z kroku 9

```

backstageIncreaseQuality(item: Item) {
  if (item.sellIn < 11) {
    this.increaseQuality(item);
  }
  if (item.sellIn < 6) {
    this.increaseQuality(item);
  }
}

updateQuality() {
  for (const item of this.items) {
    if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a
TAFKAL80ETC concert') {
      this.decreaseQuality(item);
    } else {
      this.increaseQuality(item);
      if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
        this.backstageIncreaseQuality(item);
      }
    }
    this.decreaseSellIn(item);
    if (item.sellIn < 0) {
      if (item.name !== 'Aged Brie') {
        if (item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
          this.decreaseQuality(item);
        } else {
          item.quality = item.quality - item.quality
        }
      } else {
        this.increaseQuality(item);
      }
    }
  }

  return this.items;
}

```

Krok 11: Zmiana kolejności warunków w funkcji zmieniającej parametr "Quality"


```

decreaseQuality(item: Item) {
  if (item.name !== 'Sulfuras, Hand of Ragnaros') {
    if (item.quality > 0) {
      item.quality = item.quality - 1
      if (item.name.lastIndexOf('Conjured ', 0) === 0 && item.quality > 0) {
        item.quality = item.quality - 1;
      }
    }
  }
}

```

Krok 12: Deklaracja funkcji wywoływanej przed zmniejszeniem parametru "SellIn"

```

beforeSellIn(item: Item) {
  if (item.name !== 'Aged Brie' && item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
    this.decreaseQuality(item);
  } else {
    this.increaseQuality(item);
    if (item.name === 'Backstage passes to a TAFKAL80ETC concert') {
      this.backstageIncreaseQuality(item);
    }
  }
}

```

Krok 13: Podstawienie funkcji z kroku 12

```

for (const item of this.items) {
  this.beforeSellIn(item);
  this.decreaseSellIn(item);
  if (item.sellIn < 0) {
    if (item.name !== 'Aged Brie') {
      if (item.name !== 'Backstage passes to a TAFKAL80ETC concert') {
        this.decreaseQuality(item);
      } else {
        item.quality = item.quality - item.quality
      }
    } else {
      this.increaseQuality(item);
    }
  }
}

return this.items;

```

Krok 14: Deklaracja funkcji wywoływanej po zmniejszeniu parametru "SellIn"

```

negativeSellIn(item: Item) {
    if (item.name != 'Aged Brie') {
        if (item.name != 'Backstage passes to a TAFKAL80ETC concert') {
            this.decreaseQuality(item);
        } else {
            item.quality = item.quality - item.quality
        }
    } else {
        this.increaseQuality(item);
    }
}

```

Krok 15: Podstawienie funkcji z kroku 14

```

for (const item of this.items) {
    this.beforeSellIn(item);
    this.decreaseSellIn(item);
    if (item.sellIn < 0) {
        this.negativeSellIn(item);
    }
}

return this.items;

```

Krok 16: Lista produktów, w których nie zmniejszamy parametru "Quality"

```

dontDecreaseQuality = [
    'Sulfuras, Hand of Ragnaros',
    'Backstage passes to a TAFKAL80ETC concert',
    'Aged Brie',
];

```

Krok 17: Zastosowanie tej listy w funkcji zmniejszającej parametr "Quality"

```

decreaseQuality(item: Item) {
    if (this.dontDecreaseQuality.indexOf(item.name) === -1) {
        if (item.quality > 0) {
            item.quality = item.quality - 1
            if (item.name.lastIndexOf('Conjured ', 0) === 0 && item.quality > 0) {
                item.quality = item.quality - 1;
            }
        }
    }
}

```

Krok 18: Odwrócenie warunków w funkcji zerującej wartość parametru "Quality"

```
negativeSellIn(item: Item) {  
    if (item.name == 'Aged Brie') {  
        this.increaseQuality(item);  
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {  
        item.quality = 0  
    } else {  
        this.decreaseQuality(item);  
    }  
}
```

Krok 19: Deklaracja funkcji zmniejszającej parametru "Quality" dla produktów typu "Conjured"

```
conjuredDecreaseQuality(item: Item) {  
    this.decreaseQuality(item);  
    this.decreaseQuality(item);  
}
```

Krok 20: Podstawienie funkcji z kroku 19

```

decreaseQuality(item: Item) {
    if (this.dontDecreaseQuality.indexOf(item.name) === -1) {
        if (item.quality > 0) {
            item.quality = item.quality - 1;
        }
    }
}

beforeSellIn(item: Item) {
    if (item.name != 'Aged Brie' && item.name != 'Backstage passes to a TAFKAL80ETC concert') {
        if (item.name.lastIndexOf('Conjured ', 0) === 0) {
            this.conjuredDecreaseQuality(item);
        } else {
            this.decreaseQuality(item);
        }
    } else {
        this.increaseQuality(item);
        if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
            this.backstageIncreaseQuality(item);
        }
    }
}

negativeSellIn(item: Item) {
    if (item.name == 'Aged Brie') {
        this.increaseQuality(item);
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        item.quality = 0
    } else if (item.name.lastIndexOf('Conjured ', 0) === 0) {
        this.conjuredDecreaseQuality(item);
    } else {
        this.decreaseQuality(item);
    }
}

```

Krok 21: Rearanzacja warunków w funkcji przy negatywnej wartości parametru "SellIn"

```

beforeSellIn(item: Item) {
    if (item.name.lastIndexOf('Conjured ', 0) === 0) {
        this.conjuredDecreaseQuality(item);
    } else if (item.name == 'Aged Brie') {
        this.increaseQuality(item);
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        this.increaseQuality(item);
        this.backstageIncreaseQuality(item);
    } else {
        this.decreaseQuality(item);
    }
}

```

Krok 22: Przeniesienie funkcji zwiększającej parametr "Quality" dla produktów typu "Backstage"

```

beforeSellIn(item: Item) {
    if (item.name.lastIndexOf('Conjured ', 0) === 0) {
        this.conjuredDecreaseQuality(item);
    } else if (item.name == 'Aged Brie') {
        this.increaseQuality(item);
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        this.backstageIncreaseQuality(item);
    } else {
        this.decreaseQuality(item);
    }
}

backstageIncreaseQuality(item: Item) {
    this.increaseQuality(item);
    if (item.sellIn < 11) {
        this.increaseQuality(item);
    }
    if (item.sellIn < 6) {
        this.increaseQuality(item);
    }
}

```

Kod po refaktoryzacji

```

dontDecreaseQuality = [
    'Sulfuras, Hand of Ragnaros',
    'Backstage passes to a TAFKAL80ETC concert',
    'Aged Brie'
];

constructor(items = [] as Array<Item>) {
    this.items = items;
}

```

```

}

increaseQuality(item: Item) {
    if (item.quality < 50) {
        item.quality = item.quality + 1
    }
}

conjuredDecreaseQuality(item: Item) {
    this.decreaseQuality(item);
    this.decreaseQuality(item);
}

decreaseQuality(item: Item) {
    if (this.dontDecreaseQuality.indexOf(item.name) === -1) {
        if (item.quality > 0) {
            item.quality = item.quality - 1;
        }
    }
}

backstageIncreaseQuality(item: Item) {
    this.increaseQuality(item);
    if (item.sellIn < 11) {
        this.increaseQuality(item);
    }
    if (item.sellIn < 6) {
        this.increaseQuality(item);
    }
}

decreaseSellIn(item: Item) {
    if (item.name != 'Sulfuras, Hand of Ragnaros') {
        item.sellIn = item.sellIn - 1;
    }
}

beforeSellIn(item: Item) {
    if (item.name.lastIndexOf('Conjured ', 0) === 0) {
        this.conjuredDecreaseQuality(item);
    } else if (item.name == 'Aged Brie') {
        this.increaseQuality(item);
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        this.backstageIncreaseQuality(item);
    } else {
        this.decreaseQuality(item);
    }
}

negativeSellIn(item: Item) {
    if (item.name == 'Aged Brie') {

```

```

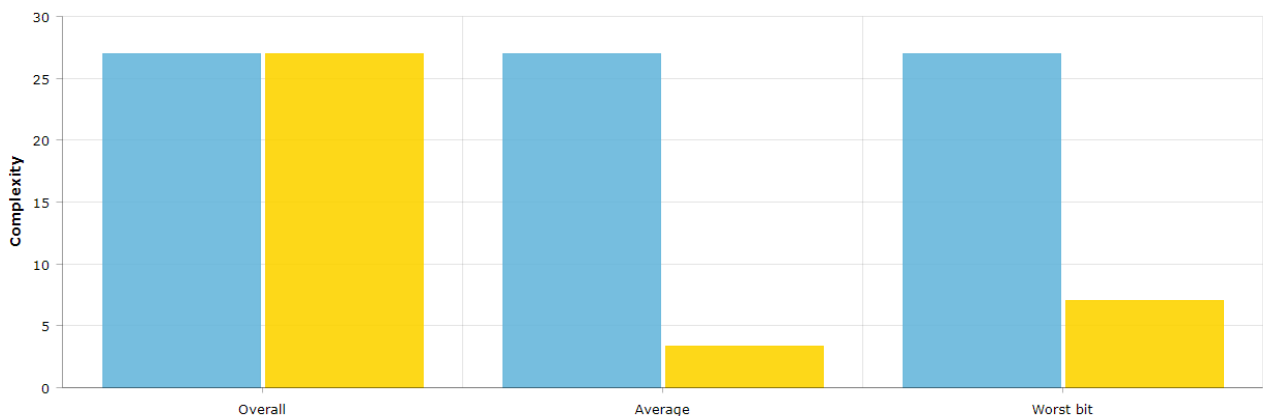
        this.increaseQuality(item);
    } else if (item.name == 'Backstage passes to a TAFKAL80ETC concert') {
        item.quality = 0
    } else if (item.name.lastIndexOf('Conjured ', 0) === 0) {
        this.conjuredDecreaseQuality(item);
    } else {
        this.decreaseQuality(item);
    }
}

updateQuality() {
    for (const item of this.items) {
        this.beforeSellIn(item);
        this.decreaseSellIn(item);
        if (item.sellIn < 0) {
            this.negativeSellIn(item);
        }
    }

    return this.items;
}

```

Wyniki złożoności



Na niebiesko pokolorowano wartości przed zmianami, a na żółto po. Można zaobserwować, że ogólna złożoność nie zmieniła się natomiast średni wynik na funkcję oraz najgorszy wynik zostały drastycznie zmniejszone.

Podsumowanie

Kod po refaktoryzacji jest bardzo podatny na zmiany i można w stosunkowo łatwy sposób dodać nowe typy produktów. Obecna wersja jest w takiej postaci, że stosunkowo łatwo można byłoby zastosować polimorfizm.