



# Code Smells Snippets

Herda Akshija

## Bloaters

Long Method

Large Class

Data Clumps

Long Parameter List

Primitive Obsession

## Object-Orientation Abusers

Switch Statements

Refused Bequest

Alternative Classes

Different Interfaces

Temporary Fields

## Change Preventers

Divergent Change

Shotgun Surgery

Parallel Inheritance

Hierarchies

## Couplers

Feature Envy

Inappropriate Intimacy

Message Chains

Middle Man

## Dispensables

Lazy class

Speculative Generality

Data Class

Duplicated Code

# Long Parameter List

```
searchHousesAsync(  
  id?: string,  
  companyNo?: string,  
  personNo?: string,  
  projectName?: string,  
  maxHits?: number  
): Promise<ResultArray<HouseDto>>;
```

# Message Chain

```
.filter((x) => filterList.includes(HouseTypes[x.houseId as HouseTypes]))
```

```
return housesToFilter  
  .filter((house) => getHouseTypes(house))
```

# Large Class

```
class Employee {  
    eat() {  
        //...  
    }  
    drink() {  
        //...  
    }  
    work() {  
        //...  
    }  
    sleep() {  
        ///...  
    }  
    walk() {  
        //...  
    }  
}
```



```
class Person {  
    eat() {  
        //...  
    }  
    drink() {  
        //...  
    }  
    sleep() {  
        ///...  
    }  
    walk() {  
        //...  
    }  
}  
  
class Employee extends Person {  
    work() {  
        //...  
    }  
}
```

# Primitive Obsession

```
const HouseCategory = 9;  
const HouseType = 'DR';  
const HouseStatus = 'SALE STATUS';
```

# Shotgun surgery

```
setActive={handleSelected} />
```

```
const HouseNo = ({ houseNo }: HouseNoProps): JSX.Element => {
  const setActiveHouseNo = useSetActiveHouseNo();
  return (
    <div className={styles.houseNo}>
      <div onClick={() => setActiveHouseNo(houseNo)} className={styles.houseNo}>
        {houseNo}
      </div>
    </div>
  );
};

export default HouseNo;
```

# Commented Code

```
//TODO
```

```
it('should return its initialState', () => {  
  // const expected = { ...initialState };  
});
```

```
/* Used for testing */  
Used for testing,
```



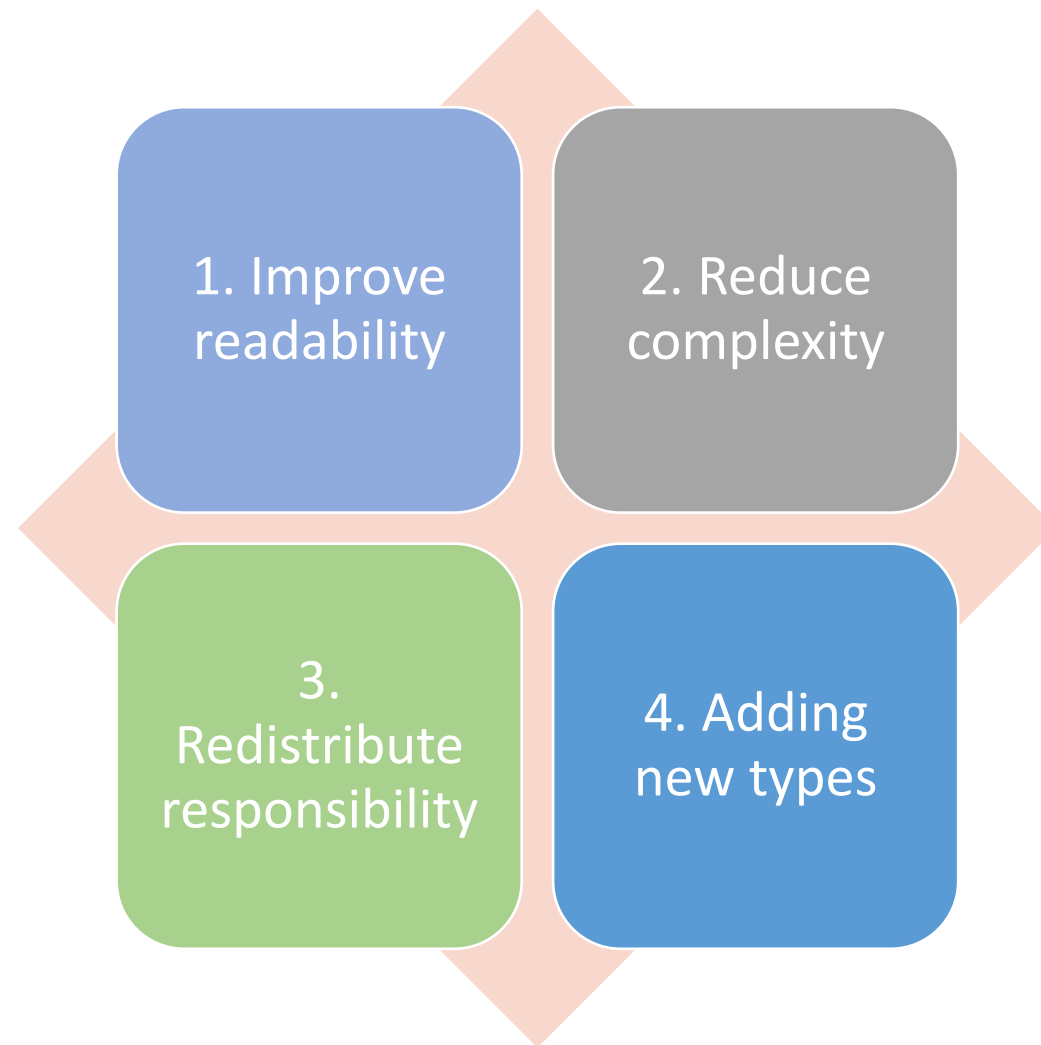
# Switch Statements

```
}: ViewHouseIconProps): JSX.Element => {  
  switch (houseCategory.toLocaleLowerCase()) {  
    case '':  
      return <ApartmentIcon aria-label={'apartment'} />;  
    case '':  
      return <CottageIcon aria-label={'cottage'} />;  
    case '':  
      return <CraftsmanHouseIcon aria-label={'craftsman'} />;  
    case '':  
      return <FarmHouseIcon aria-label={'farm'} />;  
    case '':  
      return <MediterraneanHouseIcon aria-label={'mediterranean'} />;  
    case '':  
      return <VictorianHouseIcon aria-label={'victorian'} />;  
    default:  
      return <HouseIcon aria-label={'house'} />;  
  }  
};
```

```
export enum HouseCategoryType {  
    Apartment = 'apartment',  
    Cottage = 'cottage',  
    Craftsman = 'craftsman',  
    Farm = 'farm',  
    Mediterranean = 'mediterranean',  
    Victorian = 'victorian',  
    House = 'house'  
}
```

```
function isGivenHouseCategory(houseCategoryDescription: string, houseCategoryType: HouseCategoryType) {  
    return houseCategoryDescription.includes(houseCategoryType);  
}
```

# Refactoring





# Lessons

Source:  
<https://betterprogramming.pub>  
<https://refactoring.guru/>



Questions?



# Thank You

Contact: [herda.akshija@bouvet.no](mailto:herda.akshija@bouvet.no)