Тема 14.2

Sunday, June 30, 2024 10:16 AM

SOLID принципи

1) Single responsibility - 1 компонент има тясно 1 отговорност
   - cohesion - свързаност на компонентите в класа
     Всички компоненти трябва да имат strong cohesion помежду си. В противен случай трябва да р зделим класа

```
class Triangle {

  getArea(); } strong
  getPer(); } cohesion

  draw(); } strong
  rotate(); } cohesion

};
```

weak cohesion ⟹

```
class Triangle {

  getArea();
  getPer();

};

class TriangleUI {

  draw();
  rotate();

};
```

2) Open-closed principle - Всеки клас трябва да бъде отворен за надграждане, но затворен за модификация

3) Liskov substitution principle - трябва да използваме указатели и референции от базов клас, без да се интересуваме към кой наследник са насочени

4) Interface segregation - потребителите не трябва да разчитат на интерфейс, който не използват

5) Dependency inversion - модулите от по-високо ниво на абстракция трябва да зависят от модули от по-ниско ниво

- модулите от по-високо ниво имат достъп до данните на тези от по-ниско, а тези от по-ниско работят с базата данни

- няколко зависимост от външни ресурси