

Да се напише програма за управление на робот с общ предназначение със заменяеми части.  
За изисквания за оценки прочетете т.5.

1. Клас за робот - нашите роботи ще представляват обща рамка, в която могат да се добавят различни елементи - модули. Всеки модул изпълнява дадена функция - ръце, крака и глави. Когато се сложат достатъчно модули нашият робот се счита за готов.

Класът задължително трябва да има атрибути за име, серийен номер и брой модули, които могат да се инсталират на робота. Трябва да съдържа и списък от инсталираните модули.

Да се добавят методи за:

- `void addModule(RobotModule newModule)` - добавяне на модул. Да се хвърля изключение ако подадената инстанция вече присъства в робота или е достигнат неговия максимален брой модули.
- `boolean isMinimumRequirementCovered()` - проверява дали са покрити минималните изисквания за да се счита роботът за завършен. Изискванията са:
  - има поне 1 глава
  - има поне 2 ръце
  - има поне 1 способ за задвижване
  - покрити са индивидуалните изисквания за всеки един модул(виж надолу)
- `void processCommand(String command)` - обработка команда, подадена на робота(виж надолу)

2. Клас за модул - това са различните елементи, които могат да се добавят към робота за да върши той работата си - от приготвяне на кафе до спасяване на корабкрушеници.

Всеки един модул трябва да съдържа атрибути за тип, серийен номер и уникален идентификатор. Също така да има методи:

- `String toString()` - връща полезна информация за него
- `void processCommand(String command)` - обработка команда, получена от робота(виж надолу)
- `boolean isMinimumRequirementCovered(List<RobotModule> rest)` - проверява дали са покрити изискванията на модула. За някои проверки е от значение и какви други модули присъстват

Според изискванията надолу всеки модул може да се нуждае и от допълнителни атрибути или методи

Да се имплементират и различните видове налични модули:

- глава
  - характеризира се със списък от сензори. За нашите цели е достатъчно да бъдат описани с по един стринг - камера, инфрачервен, звуков и т.н.
  - има флаг за това дали поддържа дълбочинно зрение. Той се задава автоматично от кода ако има поне 2 сензора от един и същи тип
- ръка
  - характеризира се с брой стави и максимално тегло, което може да вдигне
  - видове:
    - с пръсти
    - с щипка
- придвижване
  - характеризира се със скорост, която може да развие
  - видове:
    - крак
    - верига
    - колело
- 3. Изисквания на модулите - за метода `isMinimumRequirementCovered` трябва да се вземат предвид и индивидуалните изисквания на инсталираните модули:

| вид | разновидност | изисквания |

| - |

| глава | - | - |

| ръка | с пръсти | има поне една глава с дълбочинно зрение |

| ръка | с щипка | - |

| придвижване | крак | има поне още 1 крак и поне една глава с дълбочинно зрение |

| придвижване | верига | има поне още 1 верига |

| придвижване | колело | има поне още 2 колела |

4. Обработване на команди - роботът освен да съществува трябва да може и да върши неща. Това става като му се подават команди, които се прехвърлят към съответния модул.

Всяка команда има една от следните структури:

- <вид на модул> <действие> <аргумент> - подава се към всички модули от този вид
- <идентификатор на модул> <действие> <аргумент> - подава се само към модула с този идентификатор

Ако постъпи команда за модул, който не присъства в робота, трябва да се хвърли изключение.

Команди, които трябва да се поддържат:

| модул | команда | аргумент | действие | изключения |  
| - |  
| глава | rotate\_left | ъгъл | главата се завърта наляво с ъгъл градуса | достигнат е максимален ъгъл на завъртане |  
| глава | rotate\_right | ъгъл | главата се завърта надясно с ъгъл градуса | достигнат е максимален ъгъл на завъртане |  
| ръка | grab | тегло на предмета | ръката хваща този предмет | - |  
| ръка | release | - | ръката пуска хванатия предмет | няма хванат предмет |  
| ръка | lift | - | ръката вдига хванатия предмет | няма хванат предмет ИЛИ теглото на предмета надвишава силата на ръката |  
| крак | move\_fwd | разстояние | кракът се придвижва с разстояние единици напред | - |  
| крак | move\_back | разстояние | кракът се придвижва с разстояние единици назад | - |  
| крак | rotate\_left | ъгъл | кракът се завърта с ъгъл градуса наляво | - |  
| крак | rotate\_right | ъгъл | кракът се завърта с ъгъл градуса надясно | - |

Пример:

arm grab 100 -> всички ръце ще хванат предмета с тегло 100

arm\_7 grab 100 -> ръката с идентификатор arm\_7 ще хване предмета с тегло 100

#### 5. Допълнително:

- не е нужно да имплементирате всички модули, изброение в т.2. Достатъчно е да имате 2 от тях (например глава и ръка с пръсти)
- обработка на грешки е задължителна за оценки от 4 нагоре
- четене на команди от потребителския вход е задължително за оценки 5 и 6
- работа с потоци е задължителна за оценки 5 и 6
- частично покритие с юнит тестове е задължително за оценка 6