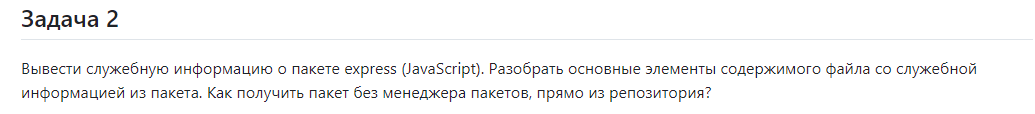


import pkg\_resources  
  
# Получение информации о пакете  
pkg\_info = pkg\_resources.get\_distribution("matplotlib").get\_metadata("METADATA")  
  
# Вывод информации о пакете  
print(pkg\_info)

git clone <https://github.com/matplotlib/matplotlib.git>



git clone https://github.com/expressjs/express.git --depth 1

const fs = require('fs');

fs.readFile('node\_modules/express/package.json', 'utf8', (err, data) => {

if (err) {

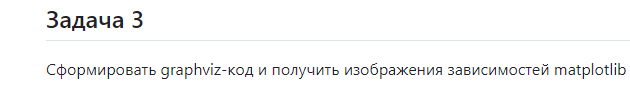
console.log('Ошибка при чтении файла:', err);

return;

}

console.log(data);

});



output\_file = f"matplotlib\_dependencies.svg"

subprocess.run(["pydeps", "matplotlib", "--noshow", "--max-bacon=2", "--output", output\_file])

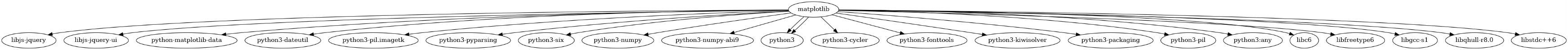
path\_to\_file = os.path.abspath(output\_file)

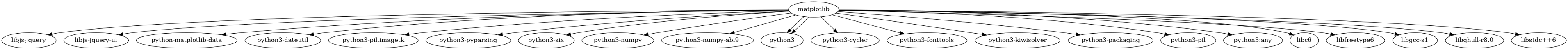
with open(path\_to\_file) as f:

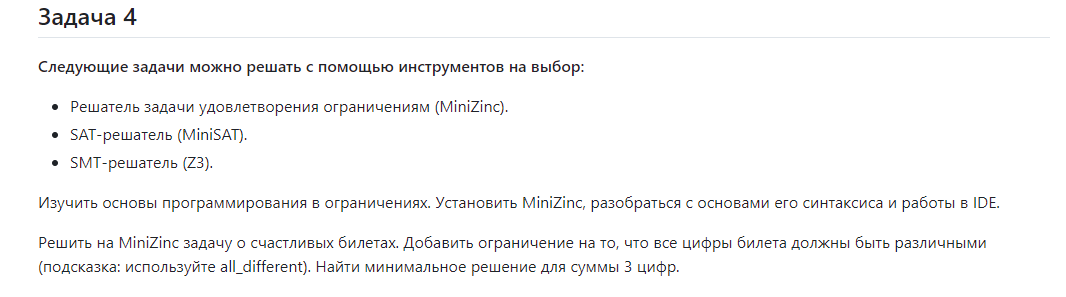
dot\_graph = f.read()

src = Source(dot\_graph)

src.render(view=True)







include *"globals.mzn"*;

var 0..9: digit1;

var 0..9: digit2;

var 0..9: digit3;

var 0..9: digit4;

var 0..9: digit5;

var 0..9: digit6;

constraint all\_different([digit1, digit2, digit3, digit4, digit5, digit6]);

constraint digit1 + digit2 + digit3 == digit4 + digit5 + digit6;

var int: sum\_group1 = digit1 + digit2 + digit3;

solve minimize sum\_group1;

output [

*"Счастливый билет с равными суммами:\n"*,

*"Первая группа: "*, show(digit1), *" + "*, show(digit2), *" + "*, show(digit3), *" = "*, show(digit1 + digit2 + digit3), *"\n"*,

*"Вторая группа: "*, show(digit4), *" + "*, show(digit5), *" + "*, show(digit6), *" = "*, show(digit4 + digit5 + digit6), *"\n"*,

*"Все цифры: "*, show([digit1, digit2, digit3, digit4, digit5, digit6])

];



array [0..1] of var 0..1: icons;

array [0..5] of var 0..1: menu;

array [0..4] of var 0..1: dropdown;

var 0..1: ic;

var 0..5: me;

var 0..4: dr;

constraint sum(icons) <= 1;

constraint sum(menu) <= 1;

constraint sum(dropdown) <= 1;

constraint icons[1] == 0;

constraint menu[0] + menu[5] >= 1;

constraint menu[0] <= dropdown[0];

constraint forall (i in 1..5) (menu[i] <= dropdown[4] + dropdown[1]);

constraint forall (i in 1..4) (dropdown[i] <= icons[1]);

constraint forall(i in 0..1) (

if icons[i] == 1

then ic=i

else true

endif

);

constraint forall(i in 0..5) (

if menu[i] == 1

then me=i

else true

endif

);

constraint forall(i in 0..4) (

if dropdown[i] == 1

then dr=i

else true

endif

);

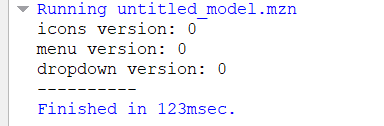
output [

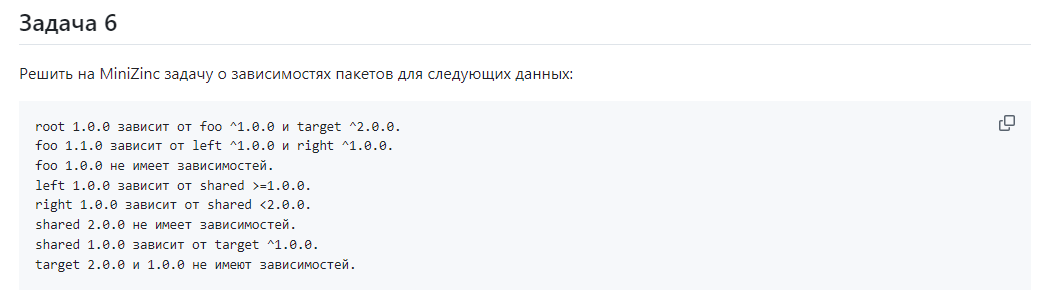
*"icons version: "*, show(ic), *"\n"*,

*"menu version: "*, show(me), *"\n"*,

*"dropdown version: "*, show(dr)

];





var 0..1: foo\_min;

var 1..9: foo\_maj;

var 0..9: target\_min;

var 1..9: target\_maj;

var 0..9: left\_min;

var 1..9: left\_maj;

var 0..9: right\_min;

var 1..9: right\_maj;

var 0..9: shared\_min;

var 1..9: shared\_maj;

constraint foo\_maj == 1;

constraint target\_maj == 2;

constraint (

if foo\_min == 1

then left\_maj = 1 /\ right\_maj = 1

else true

endif

);

constraint (

if left\_maj == 1

then shared\_maj >= 1

else true

endif

);

constraint (

if right\_maj == 1

then shared\_maj < 2

else true

endif

);

constraint (

if shared\_maj == 1

then target\_maj = 1

else true

endif

);

output [

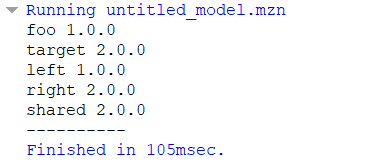
*"foo "*, show(foo\_maj), *"."*, show(foo\_min), *".0\n"*,

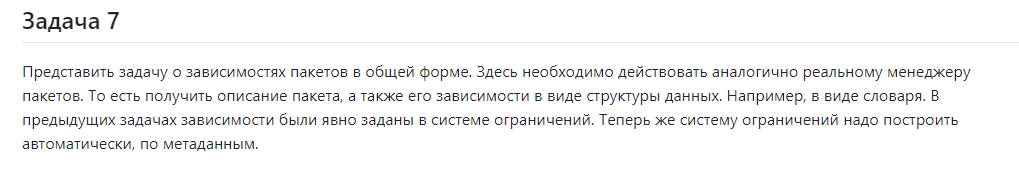
*"target "*, show(target\_maj), *"."*, show(target\_min), *".0\n"*,

*"left "*, show(left\_maj), *"."*, show(left\_min), *".0\n"*,

*"right "*, show(right\_maj), *"."*, show(right\_min), *".0\n"*,

*"shared "*, show(shared\_maj), *"."*, show(shared\_min), *".0"*

];



import requests  
  
  
def get\_package\_info(package\_name):  
 response = requests.get(f"https://pypi.org/pypi/{package\_name}/json")  
 if response.status\_code == 200:  
 data = response.json()  
 return data  
 else:  
 raise Exception(f"Не удалось получить информацию о пакете: {package\_name}")  
  
  
def parse\_dependencies(data):  
 dependencies = []  
 try:  
 releases = data['info']['requires\_dist'] or []  
 for release in releases:  
 dependencies.append(release.split(' ')[0])  
 return dependencies  
 except KeyError:  
 return dependencies  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 package\_name = input("Введите имя пакета: ")  
 package\_data = get\_package\_info(package\_name)  
 dependencies = parse\_dependencies(package\_data)  
 print(f"Зависимости для пакета {package\_name}: {dependencies}")

