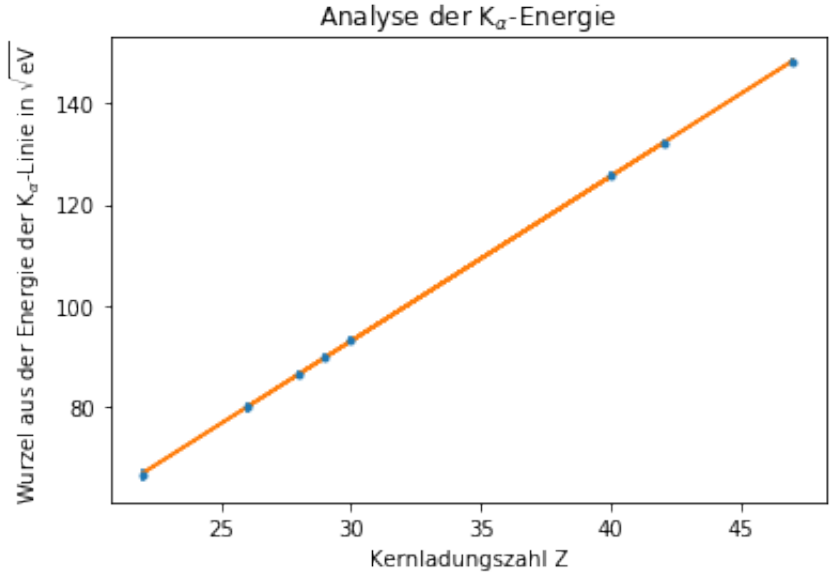
In [3]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

In [4]:
```
#data
Z = np.array([26, 42, 22, 40, 30, 47, 29, 28])
Ea = np.array([6400, 17480, 4450, 15810, 8670, 21960, 8070, 7490])
error_Ea = np.array([160, 160, 170, 170, 160, 180, 160, 160])
sq_Ea = np.sqrt(Ea)
error_sq_Ea = sq_Ea/2*error_Ea/Ea
names = np.array(['Fe', 'Mo', 'Ti', 'Zr', 'Zn', 'Ag', 'Cu', 'Ni'])

#Fit
def fit_func(z, s, Er):
    return np.sqrt(Er)*(z-s)*np.sqrt(3/4)
from scipy.optimize import curve_fit
popt, pcov=curve_fit(fit_func,Z,sq_Ea,p0=[1,13.6],sigma=error_sq_Ea)

#Plot
plt.title('Analyse der K$_\\alpha$-Energie')
plt.xlabel('Kernladungszahl Z')
plt.ylabel('Wurzel aus der Energie der K$_\\alpha$-Linie in $\\sqrt{\mathrm{eV}}$')
#plt.ylim((0,430))
plt.errorbar(Z, sq_Ea, yerr=error_sq_Ea, linestyle='None', fmt='.')
plt.plot(Z,fit_func(Z,*popt))
plt.savefig('figures/Kerladungszahlabhängigkeit_alpha.pdf',format='pdf')

#Print
chi2_= np.sum((fit_func(Z,*popt)-sq_Ea)**2/error_sq_Ea**2)
dof = 6 #degrees of freedom
chi2_red=chi2_/dof
print("chi2=",chi2_)
print("chi2_red=",chi2_red)
```

```
print("sigma=",popt[0],",Standardfehler=",np.sqrt(pcov[0][0]))
print("Er=",popt[1],",Standardfehler=",np.sqrt(pcov[1][1]))

print(' ')
print('Tabelle:')
print(' ')
print('{0:4} {1:3} {2:4} {3:3} {4:7} {5:3} {6:7} {7:3} {8:12} {9:3} {10:10}'.format( \
      'Elt.', ' | ', '  Z', ' | ', '   Ea', ' | ', '   dEa', ' | ', '   sqrt(Ea)', ' | ', 'd sqrt(Ea)'))
print('-------------------------------------------------------------------------')
for x in range(0, 7):
    print('{0:4} {1:3} {2:4d} {3:3} {4:7d} {5:3} {6:7d} {7:3} {8:12f} {9:3} {10:10f}'.format( \
          names[x], ' | ', Z[x], ' | ', Ea[x], ' | ', error_Ea[x], ' | ', sq_Ea[x], ' | ', error_sq_Ea[x
]))
```

```
chi2= 0.237121299586
chi2_red= 0.0395202165976
sigma= 1.40204983144 ,Standardfehler= 0.0756054756403
Er= 14.1172566624 ,Standardfehler= 0.0588265261434

Tabelle:

Elt.  |    Z   |     Ea    |     dEa   |     sqrt(Ea)    |  d sqrt(Ea)
-------------------------------------------------------------------------
Fe    |    26  |     6400  |      160  |     80.000000  |    1.000000
Mo    |    42  |    17480  |      160  |    132.211951  |    0.605089
Ti    |    22  |     4450  |      170  |     66.708320  |    1.274204
Zr    |    40  |    15810  |      170  |    125.737822  |    0.676010
Zn    |    30  |     8670  |      160  |     93.112835  |    0.859173
Ag    |    47  |    21960  |      180  |    148.189068  |    0.607332
Cu    |    29  |     8070  |      160  |     89.833179  |    0.890540
```

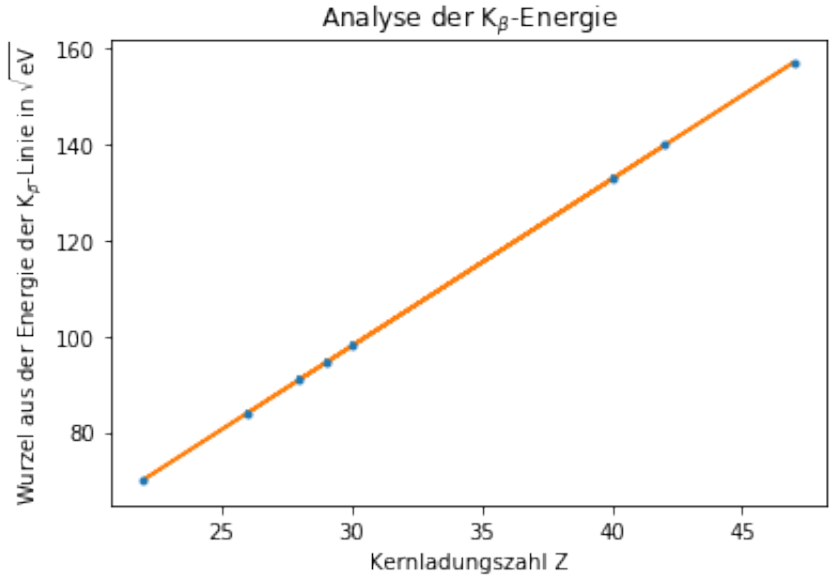


In [5]:
```
#data
Z = np.array([26, 42, 22, 40, 30, 47, 29, 28])
Eb = np.array([7050, 19600, 4890, 17700, 9620, 24580, 8950, 8290])
error_Eb = np.array([160, 150, 100, 180, 160, 160, 170, 170])
sq_Eb = np.sqrt(Eb)
error_sq_Eb = sq_Eb/2*error_Eb/Eb

#Fit
def fit_func(z, s, Er):
    return np.sqrt(Er)*(z-s)*np.sqrt(8/9)
from scipy.optimize import curve_fit
popt, pcov=curve_fit(fit_func,Z,sq_Eb,p0=[1,13.6],sigma=error_sq_Eb)

#Plot
plt.title('Analyse der K$_\\beta$-Energie')
plt.xlabel('Kernladungszahl Z')
plt.ylabel('Wurzel aus der Energie der K$_\\beta$-Linie in $\\sqrt{\mathrm{eV}}$')
```

```
#plt.ylim((0,430))
plt.errorbar(Z, sq_Eb, yerr=error_sq_Eb, linestyle='None', fmt='.')
plt.plot(Z,fit_func(Z,*popt))
plt.savefig('figures/Kerladungszahlabhängigkeit_beta.pdf',format='pdf')

#Print
chi2_= np.sum((fit_func(Z,*popt)-sq_Eb)**2/error_sq_Eb**2)
dof = 6 #degrees of freedom
chi2_red=chi2_/dof
print("chi2=",chi2_)
print("chi2_red=",chi2_red)
print("sigma=",popt[0],",Standardfehler=",np.sqrt(pcov[0][0]))
print("Er=",popt[1],",Standardfehler=",np.sqrt(pcov[1][1]))

print(' ')
print('Tabelle:')
print(' ')
print('{0:4} {1:3} {2:4} {3:3} {4:7} {5:3} {6:7} {7:3} {8:12} {9:3} {10:10}'.format( \
      'Elt.', ' | ', '  Z', ' | ', '   Eb', ' | ', '   dEb', ' | ', '   sqrt(Eb)', ' | ', 'd sqrt(Eb)'))
print('-------------------------------------------------------------------------')
for x in range(0, 7):
    print('{0:4} {1:3} {2:4d} {3:3} {4:7d} {5:3} {6:7d} {7:3} {8:12f} {9:3} {10:10f}'.format( \
          names[x], ' | ', Z[x], ' | ', Eb[x], ' | ', error_Eb[x], ' | ', sq_Eb[x], ' | ', error_sq_Eb[x
]))
```

```
chi2= 1.00359722688
chi2_red= 0.167266204481
sigma= 1.82624427841 ,Standardfehler= 0.116257700551
Er= 13.6080683942 ,Standardfehler= 0.0886199864392

Tabelle:
```

Elt.	Z	Eb	dEb	sqrt(Eb)	d sqrt(Eb)
Fe	26	7050	160	83.964278	0.952786
Mo	42	19600	150	140.000000	0.535714
Ti	22	4890	100	69.928535	0.715016
Zr	40	17700	180	133.041347	0.676481
Zn	30	9620	160	98.081599	0.815647
Ag	47	24580	160	156.780101	0.510269
Cu	29	8950	170	94.604440	0.898478



In [ ]: