In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
%matplotlib inline
```

$M = 100$ **выборок размера** $N = 1000$ **из** $R[0, \theta]$, **где** $\theta = 50, 250, 600$

1) $first \sim 2\bar{X}$

2) $second \sim (n+1)X_{(1)}$

3) $third \sim X_{(1)} + X_{(n)}$

4) $fourth \sim \frac{n+1}{n}X_{(n)}$

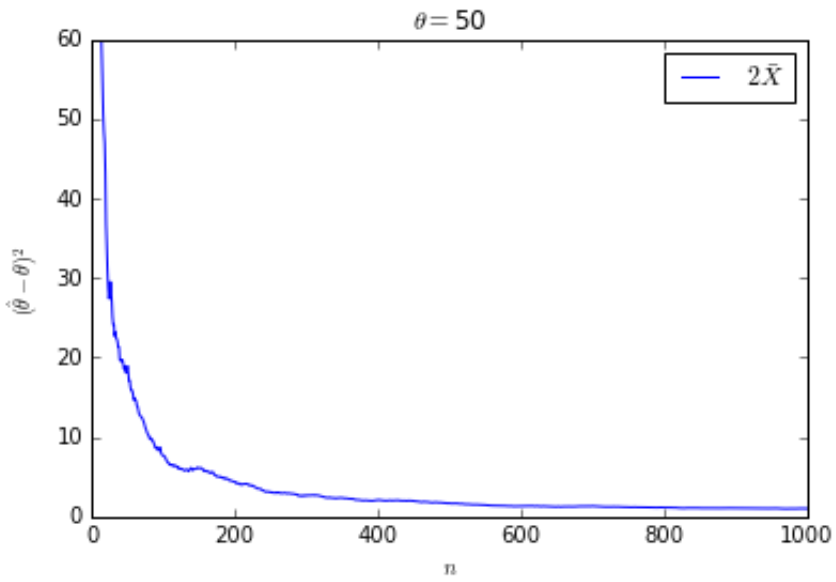
In [51]:
```
N = 1000
M = 100
theta = [50, 250, 600]
a = 0
```

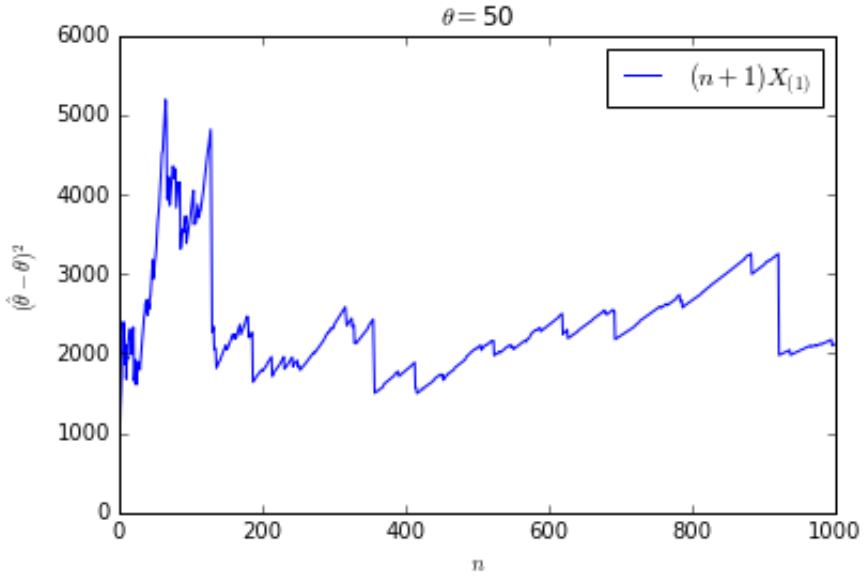
$\theta = 50$

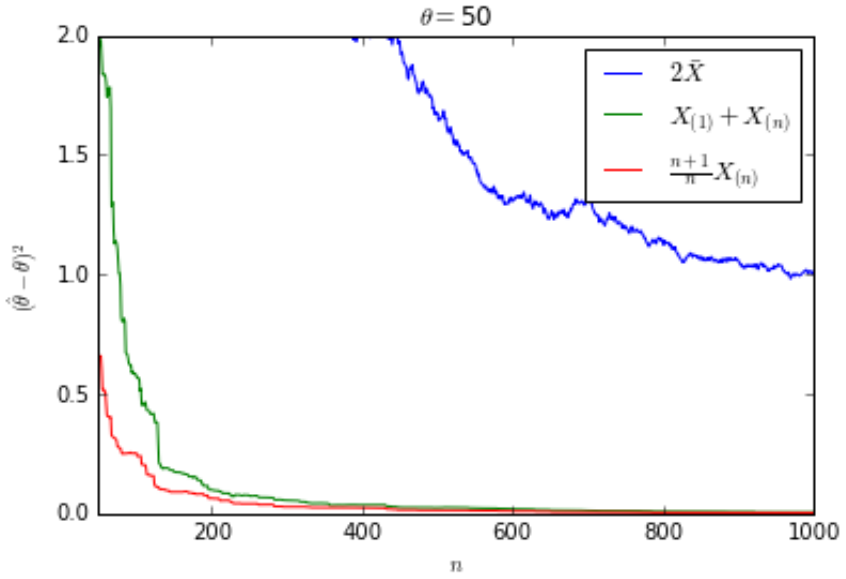
In [54]:

```
t = theta[0]
# генерируем R[0, t]
uniform_rv = sts.uniform(a, t - a)
# массивы оценок
first = np.zeros((M, N))
second = np.zeros((M, N))
third = np.zeros((M, N))
fourth = np.zeros((M, N))

for x in xrange(0,M):
    # генерируем M выборок размера N
    sample = uniform_rv.rvs(N)

    avrg = float(sample[0])
    min_el = float(sample[0])
    max_el = float(sample[0])

    first[x][0] = 2*avrg
    second[x][0] = 2*min_el
    third[x][0] = min_el + max_el
    fourth[x][0] = 2.0/1*max_el

    for n in xrange(1, N):
        # для каждого n <= N считаем оценки theta
        avrg = (avrg*n + sample[n])/(n+1)
        if(sample[n] < min_el):
            min_el = sample[n]
        if(sample[n] > max_el):
            max_el = sample[n]

        first[x][n] = 2*avrg
        second[x][n] = (n + 2)*min_el
        third[x][n] = min_el + max_el
        fourth[x][n] = (n + 2.0)/(n + 1.0)*max_el

first_plt = np.zeros(N)
second_plt = np.zeros(N)
third_plt = np.zeros(N)
fourth_plt = np.zeros(N)

# расчет разности квадрата оценки и параметра для всех оценок
for x in xrange(0,N):
    first_plt[x] = np.mean((first[:,x] - t)**2)
    second_plt[x] = np.mean((second[:,x] - t)**2)
    third_plt[x] = np.mean((third[:,x] - t)**2)
    fourth_plt[x] = np.mean((fourth[:,x] - t)**2)

# построение графиков
x = np.arange(1, N + 1, 1)
#print first_plt,x

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
```

```
plt.xlabel('$n$')
plt.legend()
plt.ylim(0,60)
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, second_plt, label = '$(n + 1)X_{(1)}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.plot(x, third_plt, label = '$X_{(1)} + X_{(n)}$')
plt.plot(x, fourth_plt, label = '$\\frac{n + 1}{n}X_{(n)}$')

plt.xlim(50,1000)
plt.ylim(0,2)
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()
```




$\theta = 50$
$(n+1)X_{(1)}$
$(\hat{\theta}-\theta)^2$
n
0
1000
2000
3000
4000
5000
6000
0
200
400
600
800
1000

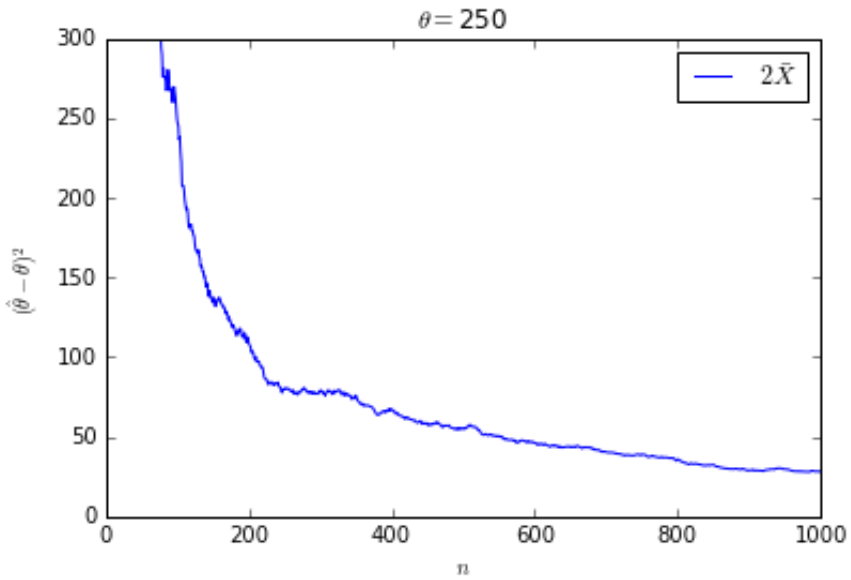

$\theta = 50$
$2\bar{X}$
$X_{(1)} + X_{(n)}$
$\frac{n+1}{n}X_{(n)}$
$(\hat{\theta}-\theta)^2$
n
0.0
0.5
1.0
1.5
2.0
200
400
600
800
1000

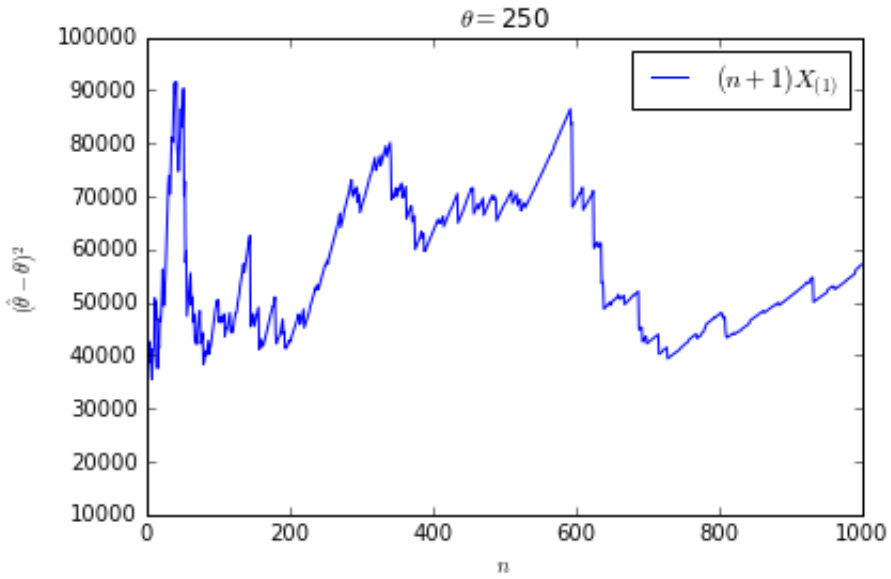
$$\theta = 250$$

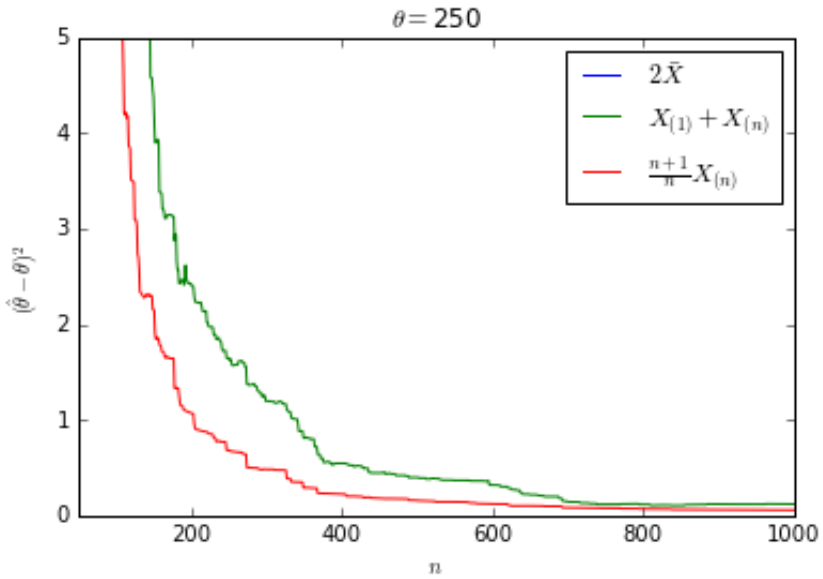
In [55]:

```
t = theta[1]
# генерируем R[0, t]
uniform_rv = sts.uniform(a, t - a)
# массивы оценок
first = np.zeros((M, N))
second = np.zeros((M, N))
third = np.zeros((M, N))
fourth = np.zeros((M, N))

for x in xrange(0,M):
    # генерируем M выборок размера N
    sample = uniform_rv.rvs(N)

    avrg = float(sample[0])
    min_el = float(sample[0])
    max_el = float(sample[0])

    first[x][0] = 2*avrg
    second[x][0] = 2*min_el
    third[x][0] = min_el + max_el
    fourth[x][0] = 2.0/1*max_el

    for n in xrange(1, N):
        # для каждого n <= N считаем оценки theta
        avrg = (avrg*n + sample[n])/(n+1)
        if(sample[n] < min_el):
            min_el = sample[n]
        if(sample[n] > max_el):
            max_el = sample[n]

        first[x][n] = 2*avrg
        second[x][n] = (n + 2)*min_el
        third[x][n] = min_el + max_el
        fourth[x][n] = (n + 2.0)/(n + 1.0)*max_el

first_plt = np.zeros(N)
second_plt = np.zeros(N)
third_plt = np.zeros(N)
fourth_plt = np.zeros(N)

# расчет разности квадрата оценки и параметра для всех оценок
for x in xrange(0,N):
    first_plt[x] = np.mean((first[:,x] - t)**2)
    second_plt[x] = np.mean((second[:,x] - t)**2)
    third_plt[x] = np.mean((third[:,x] - t)**2)
    fourth_plt[x] = np.mean((fourth[:,x] - t)**2)

# построение графиков
x = np.arange(1, N + 1, 1)
#print first_plt,x

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
```

```
plt.xlabel('$n$')
plt.legend()
plt.ylim(0,300)
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, second_plt, label = '$(n + 1)X_{(1)}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.plot(x, third_plt, label = '$X_{(1)} + X_{(n)}$')
plt.plot(x, fourth_plt, label = '$\\frac{n + 1}{n}X_{(n)}$')

plt.xlim(50,1000)
plt.ylim(0,5)
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()
```




$\theta = 250$
$(n+1)X_{(1)}$
$(\hat{\theta}-\theta)^2$
n

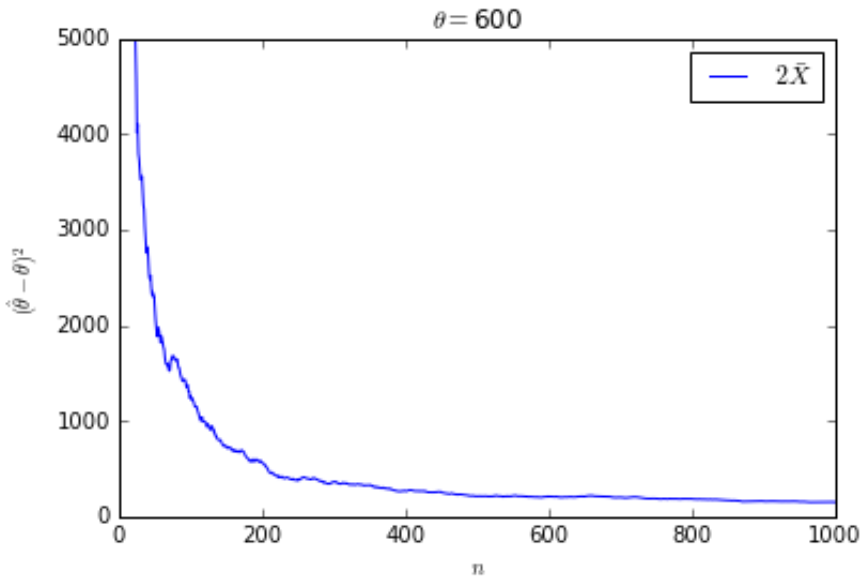

$\theta = 250$
$2\bar{X}$
$X_{(1)} + X_{(n)}$
$\frac{n+1}{n}X_{(n)}$
$(\hat{\theta}-\theta)^2$
n

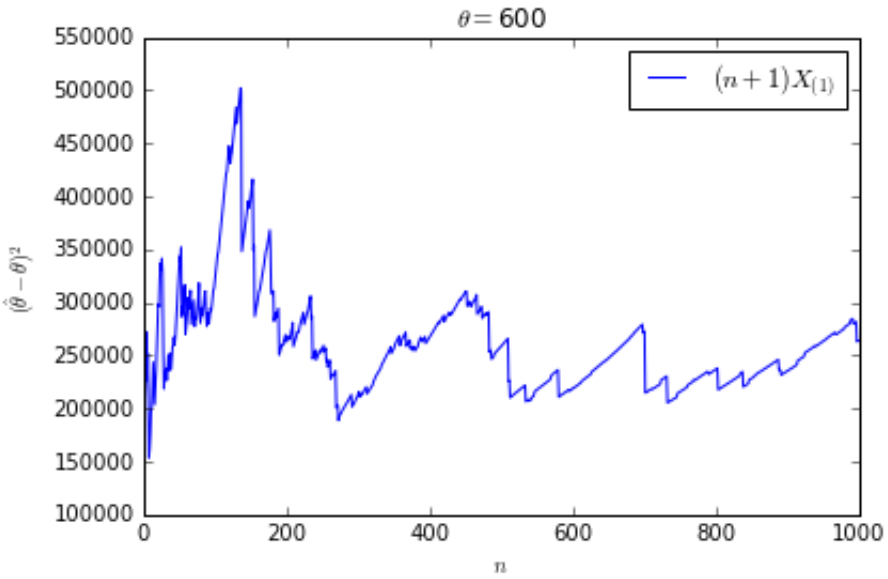
$$\theta = 600$$

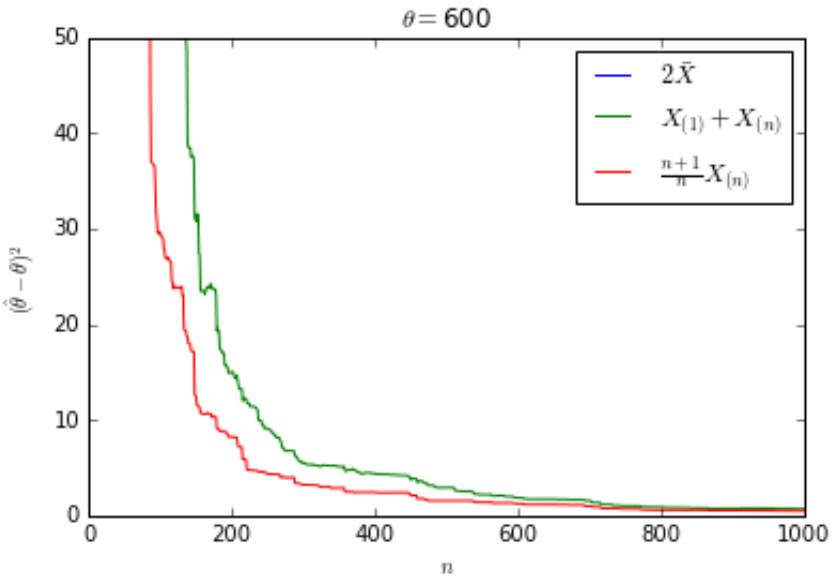
In [50]:

```
t = theta[2]
# генерируем R[0, t]
uniform_rv = sts.uniform(a, t - a)
# массивы оценок
first = np.zeros((M, N))
second = np.zeros((M, N))
third = np.zeros((M, N))
fourth = np.zeros((M, N))

for x in xrange(0,M):
    # генерируем M выборок размера N
    sample = uniform_rv.rvs(N)

    avrg = float(sample[0])
    min_el = float(sample[0])
    max_el = float(sample[0])

    first[x][0] = 2*avrg
    second[x][0] = 2*min_el
    third[x][0] = min_el + max_el
    fourth[x][0] = 2.0/1*max_el

    for n in xrange(1, N):
        # для каждого n <= N считаем оценки theta
        avrg = (avrg*n + sample[n])/(n+1)
        if(sample[n] < min_el):
            min_el = sample[n]
        if(sample[n] > max_el):
            max_el = sample[n]

        first[x][n] = 2*avrg
        second[x][n] = (n + 2)*min_el
        third[x][n] = min_el + max_el
        fourth[x][n] = (n + 2.0)/(n + 1.0)*max_el

first_plt = np.zeros(N)
second_plt = np.zeros(N)
third_plt = np.zeros(N)
fourth_plt = np.zeros(N)

# расчет разности квадрата оценки и параметра для всех оценок
for x in xrange(0,N):
    first_plt[x] = np.mean((first[:,x] - t)**2)
    second_plt[x] = np.mean((second[:,x] - t)**2)
    third_plt[x] = np.mean((third[:,x] - t)**2)
    fourth_plt[x] = np.mean((fourth[:,x] - t)**2)

# построение графиков
x = np.arange(1, N + 1, 1)
#print first_plt,x

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
```

```
plt.xlabel('$n$')
plt.legend()
plt.ylim(0,5000)
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, second_plt, label = '$(n + 1)X_{(1)}$')
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()

plt.plot(x, first_plt, label = '$2\\bar{X}$')
plt.plot(x, third_plt, label = '$X_{(1)} + X_{(n)}$')
plt.plot(x, fourth_plt, label = '$\\frac{n + 1}{n}X_{(n)}$')

plt.ylim(0,50)
plt.ylabel('$(\\hat{\\theta} - \\theta)^{2}$')
plt.xlabel('$n$')
plt.legend()
plt.title("$\\theta = $"+str(t))
plt.show()
```




$\theta = 600$
$(n+1)X_{(1)}$
$(\hat{\theta}-\theta)^2$
n


$\theta = 600$
$2\bar{X}$
$X_{(1)} + X_{(n)}$
$\frac{n+1}{n}X_{(n)}$
$(\hat{\theta}-\theta)^2$
n