In [1]:

```
from keras.applications import VGG16

conv_base = VGG16(weights='imagenet', include_top=False,
                  input_shape = (150,150,3))

print(conv_base.summary())
```

```
Using TensorFlow backend.

WARNING:tensorflow:From C:\ProgramData\Anaconda3\lib\site-packages\tensorflow\pyth
on\framework\op_def_library.py:263: colocate_with (from tensorflow.python.framewor
k.ops) is deprecated and will be removed in a future version.
Instructions for updating:
Colocations handled automatically by placer.
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 150, 150, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 150, 150, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 150, 150, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 75, 75, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 75, 75, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 75, 75, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 37, 37, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 37, 37, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 18, 18, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 18, 18, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 9, 9, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
None
```

In [3]:

```
import os
import numpy as np
from keras.preprocessing.image import ImageDataGenerator

base_dir = './datasets/cats_and_dogs_small'
train_dir = os.path.join(base_dir, 'train')
validation_dir = os.path.join(base_dir, 'validation')
test_dir = os.path.join(base_dir, 'test')

datagen = ImageDataGenerator(rescale=1./255)
batch_size = 20

def extract_features(directory, sample_count):
    features = np.zeros(shape=(sample_count, 4,4,512))
    labels = np.zeros(shape=(sample_count))
    generator = datagen.flow_from_directory(
        directory,
        target_size=(150,150),
        batch_size=batch_size,
        class_mode='binary'
    )
    i = 0
    for inputs_batch, labels_batch in generator:
        features_batch = conv_base.predict(inputs_batch)
        features[i * batch_size : (i + 1) * batch_size] = features_batch
        labels[i * batch_size : (i + 1) * batch_size] = labels_batch
        i += 1
        if i * batch_size >= sample_count:
            break
        return features, labels

train_features, train_labels = extract_features(train_dir, 2000)
validation_features, validation_labels = extract_features(validation_dir, 1000)
test_features, test_labels = extract_features(test_dir, 1000)
```

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
```

In [6]:

```
train_features =np.reshape(train_features, (2000, 4 * 4 * 512))
validation_features = np.reshape(validation_features, (1000, 4 * 4 * 512))
test_features = np.reshape(test_features, (1000, 4 * 4 * 512))
```

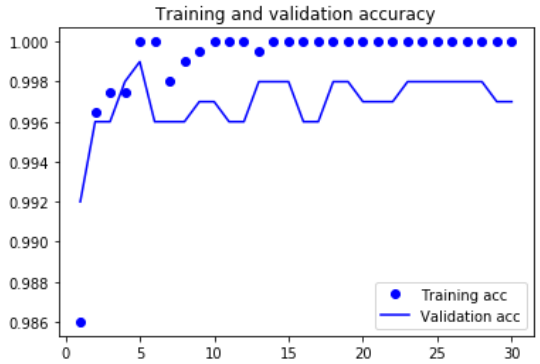
In [7]:

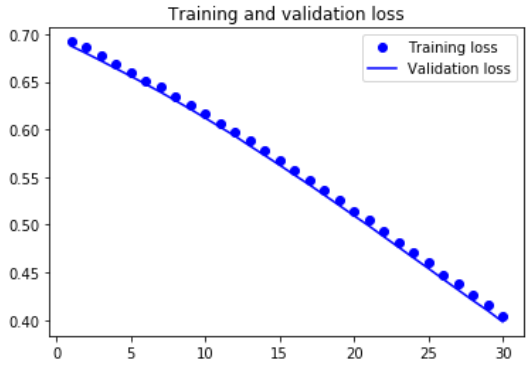
```
from keras import  models, layers, optimizers

model = models.Sequential()
model.add(layers.Dense(256, activation='relu', input_dim=4 * 4 * 512))
model.add(layers.Dropout(0.5))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer=optimizers.RMSprop(lr=2e-5),
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit(
    train_features, train_labels,
    epochs=30,
    batch_size=20,
    validation_data = (validation_features, validation_labels))
```

```
WARNING:tensorflow:From C:₩ProgramData₩Anaconda3₩lib₩site-packages₩tensorflow₩pyth
on₩ops₩math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is depreca
ted and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
Train on 2000 samples, validate on 1000 samples
Epoch 1/30
2000/2000 [==============================] - 1s 691us/step - loss: 0.6920 - acc:
0.9860 - val_loss: 0.6875 - val_acc: 0.9920
Epoch 2/30
2000/2000 [==============================] - 1s 288us/step - loss: 0.6869 - acc:
0.9965 - val_loss: 0.6799 - val_acc: 0.9960
Epoch 3/30
2000/2000 [==============================] - 1s 289us/step - loss: 0.6772 - acc:
0.9975 - val_loss: 0.6720 - val_acc: 0.9960
Epoch 4/30
2000/2000 [==============================] - 1s 292us/step - loss: 0.6692 - acc:
0.9975 - val_loss: 0.6639 - val_acc: 0.9980
Epoch 5/30
2000/2000 [==============================] - 1s 286us/step - loss: 0.6596 - acc:
1.0000 - val_loss: 0.6559 - val_acc: 0.9990
Epoch 6/30
2000/2000 [==============================] - 1s 290us/step - loss: 0.6508 - acc:
1.0000 - val_loss: 0.6475 - val_acc: 0.9960
Epoch 7/30
2000/2000 [==============================] - 1s 288us/step - loss: 0.6443 - acc:
0.9980 - val_loss: 0.6391 - val_acc: 0.9960
Epoch 8/30
2000/2000 [==============================] - 1s 296us/step - loss: 0.6345 - acc:
0.9990 - val_loss: 0.6303 - val_acc: 0.9960
Epoch 9/30
2000/2000 [==============================] - 1s 344us/step - loss: 0.6254 - acc:
0.9995 - val_loss: 0.6213 - val_acc: 0.9970
Epoch 10/30
2000/2000 [==============================] - 1s 343us/step - loss: 0.6162 - acc:
1.0000 - val_loss: 0.6120 - val_acc: 0.9970
Epoch 11/30
2000/2000 [==============================] - 1s 343us/step - loss: 0.6068 - acc:
1.0000 - val_loss: 0.6026 - val_acc: 0.9960
Epoch 12/30
2000/2000 [==============================] - 1s 344us/step - loss: 0.5973 - acc:
1.0000 - val_loss: 0.5931 - val_acc: 0.9960
Epoch 13/30
2000/2000 [==============================] - 1s 347us/step - loss: 0.5883 - acc:
0.9995 - val_loss: 0.5827 - val_acc: 0.9980
Epoch 14/30
2000/2000 [==============================] - 1s 314us/step - loss: 0.5779 - acc:
1.0000 - val_loss: 0.5728 - val_acc: 0.9980
Epoch 15/30
2000/2000 [==============================] - 1s 310us/step - loss: 0.5674 - acc:
1.0000 - val_loss: 0.5626 - val_acc: 0.9980
Epoch 16/30
2000/2000 [==============================] - 1s 305us/step - loss: 0.5574 - acc:
1.0000 - val_loss: 0.5522 - val_acc: 0.9960
Epoch 17/30
2000/2000 [==============================] - 1s 296us/step - loss: 0.5474 - acc:
1.0000 - val_loss: 0.5420 - val_acc: 0.9960
Epoch 18/30
2000/2000 [==============================] - 1s 292us/step - loss: 0.5358 - acc:
1.0000 - val_loss: 0.5312 - val_acc: 0.9980
Epoch 19/30
2000/2000 [==============================] - 1s 288us/step - loss: 0.5259 - acc:
1.0000 - val_loss: 0.5204 - val_acc: 0.9980
Epoch 20/30
2000/2000 [==============================] - 1s 296us/step - loss: 0.5144 - acc:
1.0000 - val_loss: 0.5095 - val_acc: 0.9970
Epoch 21/30
2000/2000 [==============================] - 1s 285us/step - loss: 0.5044 - acc:
1.0000 - val_loss: 0.4989 - val_acc: 0.9970
Epoch 22/30
2000/2000 [==============================] - 1s 284us/step - loss: 0.4934 - acc:
1.0000 - val_loss: 0.4876 - val_acc: 0.9970
Epoch 23/30
2000/2000 [==============================] - 1s 285us/step - loss: 0.4820 - acc:
1.0000 - val_loss: 0.4765 - val_acc: 0.9980
Epoch 24/30
2000/2000 [==============================] - 1s 285us/step - loss: 0.4708 - acc:
1.0000 - val_loss: 0.4653 - val_acc: 0.9980
Epoch 25/30
2000/2000 [==============================] - 1s 301us/step - loss: 0.4600 - acc:
1.0000 - val_loss: 0.4543 - val_acc: 0.9980
Epoch 26/30
2000/2000 [==============================] - 1s 292us/step - loss: 0.4476 - acc:
1.0000 - val_loss: 0.4429 - val_acc: 0.9980
Epoch 27/30
2000/2000 [==============================] - 1s 297us/step - loss: 0.4375 - acc:
1.0000 - val_loss: 0.4317 - val_acc: 0.9980
Epoch 28/30
2000/2000 [==============================] - 1s 296us/step - loss: 0.4264 - acc:
1.0000 - val_loss: 0.4205 - val_acc: 0.9980
Epoch 29/30
2000/2000 [==============================] - 1s 302us/step - loss: 0.4158 - acc:
1.0000 - val_loss: 0.4098 - val_acc: 0.9970
Epoch 30/30
2000/2000 [==============================] - 1s 312us/step - loss: 0.4040 - acc:
1.0000 - val_loss: 0.3982 - val_acc: 0.9970
```

In [9]:

```
import matplotlib.pyplot as plt

acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(acc) + 1)

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





with 데이터 증식

In [11]:

```
from keras import  models, layers, optimizers

model = models.Sequential()
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

In [12]:

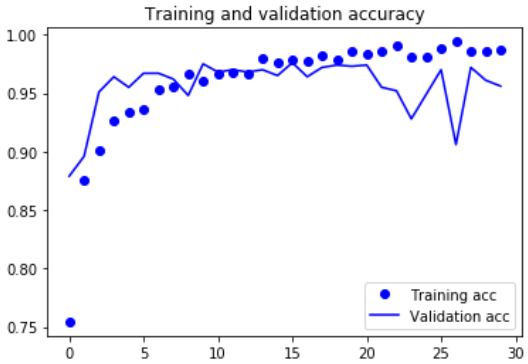
```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode = 'nearest')

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150,150),
    batch_size=20,
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(150,150),
    batch_size=20,
    class_mode='binary')

model.compile(optimizer=optimizers.RMSprop(lr=2e-5),
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=30,
    validation_data = validation_generator,
    validation_steps=50,
    verbose=2)
```

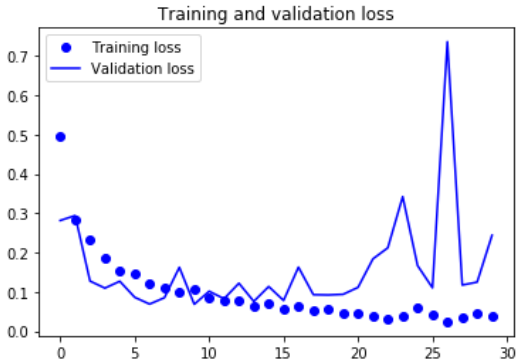
```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Epoch 1/30
 - 17s - loss: 0.4951 - acc: 0.7545 - val_loss: 0.2814 - val_acc: 0.8790
Epoch 2/30
 - 15s - loss: 0.2813 - acc: 0.8750 - val_loss: 0.2938 - val_acc: 0.8960
Epoch 3/30
 - 15s - loss: 0.2328 - acc: 0.9015 - val_loss: 0.1273 - val_acc: 0.9510
Epoch 4/30
 - 15s - loss: 0.1875 - acc: 0.9265 - val_loss: 0.1094 - val_acc: 0.9640
Epoch 5/30
 - 16s - loss: 0.1530 - acc: 0.9340 - val_loss: 0.1269 - val_acc: 0.9550
Epoch 6/30
 - 16s - loss: 0.1458 - acc: 0.9365 - val_loss: 0.0861 - val_acc: 0.9670
Epoch 7/30
 - 15s - loss: 0.1196 - acc: 0.9535 - val_loss: 0.0687 - val_acc: 0.9670
Epoch 8/30
 - 15s - loss: 0.1108 - acc: 0.9560 - val_loss: 0.0851 - val_acc: 0.9620
Epoch 9/30
 - 15s - loss: 0.0978 - acc: 0.9670 - val_loss: 0.1623 - val_acc: 0.9480
Epoch 10/30
 - 16s - loss: 0.1058 - acc: 0.9605 - val_loss: 0.0683 - val_acc: 0.9750
Epoch 11/30
 - 15s - loss: 0.0832 - acc: 0.9670 - val_loss: 0.1014 - val_acc: 0.9680
Epoch 12/30
 - 15s - loss: 0.0779 - acc: 0.9680 - val_loss: 0.0824 - val_acc: 0.9700
Epoch 13/30
 - 15s - loss: 0.0783 - acc: 0.9665 - val_loss: 0.1218 - val_acc: 0.9680
Epoch 14/30
 - 15s - loss: 0.0625 - acc: 0.9795 - val_loss: 0.0756 - val_acc: 0.9700
Epoch 15/30
 - 15s - loss: 0.0698 - acc: 0.9765 - val_loss: 0.1134 - val_acc: 0.9650
Epoch 16/30
 - 15s - loss: 0.0573 - acc: 0.9790 - val_loss: 0.0780 - val_acc: 0.9760
Epoch 17/30
 - 15s - loss: 0.0623 - acc: 0.9775 - val_loss: 0.1626 - val_acc: 0.9640
Epoch 18/30
 - 15s - loss: 0.0522 - acc: 0.9820 - val_loss: 0.0927 - val_acc: 0.9720
Epoch 19/30
 - 15s - loss: 0.0548 - acc: 0.9790 - val_loss: 0.0921 - val_acc: 0.9740
Epoch 20/30
 - 15s - loss: 0.0458 - acc: 0.9860 - val_loss: 0.0936 - val_acc: 0.9730
Epoch 21/30
 - 15s - loss: 0.0460 - acc: 0.9830 - val_loss: 0.1113 - val_acc: 0.9740
Epoch 22/30
 - 16s - loss: 0.0381 - acc: 0.9855 - val_loss: 0.1837 - val_acc: 0.9550
Epoch 23/30
 - 16s - loss: 0.0316 - acc: 0.9910 - val_loss: 0.2119 - val_acc: 0.9520
Epoch 24/30
 - 15s - loss: 0.0396 - acc: 0.9810 - val_loss: 0.3419 - val_acc: 0.9280
Epoch 25/30
 - 15s - loss: 0.0600 - acc: 0.9815 - val_loss: 0.1665 - val_acc: 0.9490
Epoch 26/30
 - 16s - loss: 0.0400 - acc: 0.9885 - val_loss: 0.1102 - val_acc: 0.9700
Epoch 27/30
 - 15s - loss: 0.0240 - acc: 0.9940 - val_loss: 0.7360 - val_acc: 0.9060
Epoch 28/30
 - 15s - loss: 0.0358 - acc: 0.9855 - val_loss: 0.1173 - val_acc: 0.9720
Epoch 29/30
 - 16s - loss: 0.0469 - acc: 0.9855 - val_loss: 0.1247 - val_acc: 0.9610
```

```
Epoch 30/30
 - 16s - loss: 0.0394 - acc: 0.9875 - val_loss: 0.2437 - val_acc: 0.9560
```

In [13]:

```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





In [14]:

```
conv_base.trainable = True

set_trainable = False
for layer in conv_base.layers:
    if layer.name == 'block5_conv1':
        set_trainable = True
    if set_trainable:
        layer.trainable = True
    else:
        layer.trainable = False
```

In [15]:

```
model.compile(optimizer=optimizers.RMSprop(lr=1e-5),
              loss='binary_crossentropy',
              metrics=['acc'])

history = model.fit_generator(
    train_generator,
    steps_per_epoch=100,
    epochs=100,
    validation_data = validation_generator,
    validation_steps=50)
```

```
Epoch 1/100
100/100 [==============================] - 17s 173ms/step - loss: 0.0235 - acc: 0.
9925 - val_loss: 0.1237 - val_acc: 0.9770
Epoch 2/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0122 - acc: 0.
9960 - val_loss: 0.1425 - val_acc: 0.9730
Epoch 3/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0102 - acc: 0.
9955 - val_loss: 0.1564 - val_acc: 0.9720
Epoch 4/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0177 - acc: 0.
9955 - val_loss: 0.1502 - val_acc: 0.9740
Epoch 5/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0139 - acc: 0.
9955 - val_loss: 0.1678 - val_acc: 0.9740
Epoch 6/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0133 - acc: 0.
9940 - val_loss: 0.1344 - val_acc: 0.9750
Epoch 7/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0099 - acc: 0.
9965 - val_loss: 0.1405 - val_acc: 0.9770
Epoch 8/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0097 - acc: 0.
9970 - val_loss: 0.1514 - val_acc: 0.9790
Epoch 9/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0076 - acc: 0.
9985 - val_loss: 0.1483 - val_acc: 0.9780
Epoch 10/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0056 - acc: 0.
9985 - val_loss: 0.1609 - val_acc: 0.9780
Epoch 11/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0091 - acc: 0.
9970 - val_loss: 0.1717 - val_acc: 0.9700
Epoch 12/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0069 - acc: 0.
9975 - val_loss: 0.1694 - val_acc: 0.9770
Epoch 13/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0095 - acc: 0.
9970 - val_loss: 0.1832 - val_acc: 0.9710
Epoch 14/100
100/100 [==============================] - 16s 158ms/step - loss: 0.0072 - acc: 0.
9970 - val_loss: 0.1512 - val_acc: 0.9800
Epoch 15/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0109 - acc: 0.
9970 - val_loss: 0.1604 - val_acc: 0.9770
Epoch 16/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0173 - acc: 0.
9960 - val_loss: 0.1660 - val_acc: 0.9760
Epoch 17/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0103 - acc: 0.
9955 - val_loss: 0.1404 - val_acc: 0.9820
Epoch 18/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0117 - acc: 0.
9960 - val_loss: 0.1502 - val_acc: 0.9790
Epoch 19/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0075 - acc: 0.
9985 - val_loss: 0.1551 - val_acc: 0.9780
Epoch 20/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0077 - acc: 0.
9975 - val_loss: 0.1549 - val_acc: 0.9780
Epoch 21/100
```

```
100/100 [==============================] - 16s 156ms/step - loss: 0.0043 - acc: 0.
9990 - val_loss: 0.1770 - val_acc: 0.9790
Epoch 22/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0035 - acc: 0.
9980 - val_loss: 0.1836 - val_acc: 0.9780
Epoch 23/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0066 - acc: 0.
9975 - val_loss: 0.1738 - val_acc: 0.9780
Epoch 24/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0025 - acc: 0.
9990 - val_loss: 0.1827 - val_acc: 0.9770
Epoch 25/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0049 - acc: 0.
9980 - val_loss: 0.1878 - val_acc: 0.9760
Epoch 26/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0076 - acc: 0.
9985 - val_loss: 0.1846 - val_acc: 0.9800
Epoch 27/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0096 - acc: 0.
9975 - val_loss: 0.1756 - val_acc: 0.9790
Epoch 28/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0031 - acc: 0.
9990 - val_loss: 0.2099 - val_acc: 0.9750
Epoch 29/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0025 - acc: 0.
9990 - val_loss: 0.1936 - val_acc: 0.9780
Epoch 30/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0021 - acc: 0.
9990 - val_loss: 0.2100 - val_acc: 0.9740
Epoch 31/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0042 - acc: 0.
9990 - val_loss: 0.2008 - val_acc: 0.9750
Epoch 32/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0055 - acc: 0.
9975 - val_loss: 0.2226 - val_acc: 0.9720
Epoch 33/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0067 - acc: 0.
9970 - val_loss: 0.1972 - val_acc: 0.9780
Epoch 34/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0058 - acc: 0.
9985 - val_loss: 0.2003 - val_acc: 0.9780
Epoch 35/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0023 - acc: 0.
9990 - val_loss: 0.1808 - val_acc: 0.9780
Epoch 36/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0169 - acc: 0.
9950 - val_loss: 0.1974 - val_acc: 0.9750
Epoch 37/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0047 - acc: 0.
9990 - val_loss: 0.2074 - val_acc: 0.9740
Epoch 38/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0069 - acc: 0.
9985 - val_loss: 0.1907 - val_acc: 0.9770
Epoch 39/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0072 - acc: 0.
9990 - val_loss: 0.2030 - val_acc: 0.9770
Epoch 40/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0085 - acc: 0.
9975 - val_loss: 0.2477 - val_acc: 0.9690
Epoch 41/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0030 - acc: 0.
9985 - val_loss: 0.2009 - val_acc: 0.9740
Epoch 42/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0122 - acc: 0.
9975 - val_loss: 0.2136 - val_acc: 0.9720
Epoch 43/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0028 - acc: 0.
9985 - val_loss: 0.1989 - val_acc: 0.9740
Epoch 44/100
100/100 [==============================] - 15s 155ms/step - loss: 7.4041e-04 - ac
c: 1.0000 - val_loss: 0.2075 - val_acc: 0.9800
Epoch 45/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0107 - acc: 0.
9965 - val_loss: 0.2178 - val_acc: 0.9740
Epoch 46/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0030 - acc: 0.
9975 - val_loss: 0.1962 - val_acc: 0.9760
Epoch 47/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0030 - acc: 0.
9990 - val_loss: 0.2232 - val_acc: 0.9790
Epoch 48/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0031 - acc: 0.
9995 - val_loss: 0.2124 - val_acc: 0.9800
Epoch 49/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0023 - acc: 0.
9990 - val_loss: 0.2192 - val_acc: 0.9780
Epoch 50/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0023 - acc: 0.
9985 - val_loss: 0.2419 - val_acc: 0.9740
Epoch 51/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0076 - acc: 0.
9980 - val_loss: 0.2202 - val_acc: 0.9740
Epoch 52/100
100/100 [==============================] - 16s 159ms/step - loss: 0.0027 - acc: 0.
9990 - val_loss: 0.2196 - val_acc: 0.9780
Epoch 53/100
100/100 [==============================] - 16s 160ms/step - loss: 0.0022 - acc: 0.
9990 - val_loss: 0.2347 - val_acc: 0.9740
Epoch 54/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0058 - acc: 0.
9980 - val_loss: 0.2327 - val_acc: 0.9740
Epoch 55/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0022 - acc: 0.
9995 - val_loss: 0.2065 - val_acc: 0.9790
Epoch 56/100
100/100 [==============================] - 16s 158ms/step - loss: 0.0027 - acc: 0.
9990 - val_loss: 0.2233 - val_acc: 0.9750
Epoch 57/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0066 - acc: 0.
9985 - val_loss: 0.1994 - val_acc: 0.9790
Epoch 58/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0025 - acc: 0.
9985 - val_loss: 0.2310 - val_acc: 0.9740
Epoch 59/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0052 - acc: 0.
9985 - val_loss: 0.2726 - val_acc: 0.9740
Epoch 60/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0031 - acc: 0.
9985 - val_loss: 0.2370 - val_acc: 0.9740
Epoch 61/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0018 - acc: 0.
9995 - val_loss: 0.2701 - val_acc: 0.9720
```

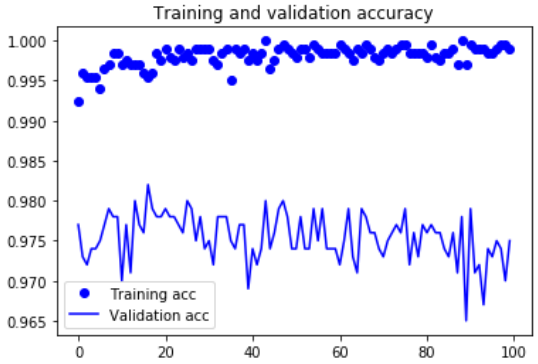
```
Epoch 62/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0025 - acc: 0.
9990 - val_loss: 0.2528 - val_acc: 0.9750
Epoch 63/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0090 - acc: 0.
9985 - val_loss: 0.2032 - val_acc: 0.9790
Epoch 64/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0043 - acc: 0.
9975 - val_loss: 0.2296 - val_acc: 0.9730
Epoch 65/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0022 - acc: 0.
9990 - val_loss: 0.2577 - val_acc: 0.9710
Epoch 66/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0060 - acc: 0.
9985 - val_loss: 0.2172 - val_acc: 0.9790
Epoch 67/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0013 - acc: 0.
9995 - val_loss: 0.2322 - val_acc: 0.9780
Epoch 68/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0041 - acc: 0.
9990 - val_loss: 0.2539 - val_acc: 0.9760
Epoch 69/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0026 - acc: 0.
9980 - val_loss: 0.2575 - val_acc: 0.9760
Epoch 70/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0080 - acc: 0.
9975 - val_loss: 0.2348 - val_acc: 0.9740
Epoch 71/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0025 - acc: 0.
9985 - val_loss: 0.2581 - val_acc: 0.9730
Epoch 72/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0084 - acc: 0.
9990 - val_loss: 0.2375 - val_acc: 0.9750
Epoch 73/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0076 - acc: 0.
9985 - val_loss: 0.2259 - val_acc: 0.9760
Epoch 74/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0017 - acc: 0.
9990 - val_loss: 0.2203 - val_acc: 0.9770
Epoch 75/100
100/100 [==============================] - 16s 156ms/step - loss: 6.5926e-04 - ac
c: 0.9995 - val_loss: 0.2341 - val_acc: 0.9760
Epoch 76/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0029 - acc: 0.
9995 - val_loss: 0.2103 - val_acc: 0.9790
Epoch 77/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0042 - acc: 0.
9985 - val_loss: 0.2352 - val_acc: 0.9720
Epoch 78/100
100/100 [==============================] - 16s 157ms/step - loss: 0.0123 - acc: 0.
9985 - val_loss: 0.2283 - val_acc: 0.9760
Epoch 79/100
100/100 [==============================] - 15s 154ms/step - loss: 0.0070 - acc: 0.
9985 - val_loss: 0.2443 - val_acc: 0.9730
Epoch 80/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0059 - acc: 0.
9985 - val_loss: 0.2129 - val_acc: 0.9770
Epoch 81/100
100/100 [==============================] - 15s 154ms/step - loss: 0.0099 - acc: 0.
9980 - val_loss: 0.2206 - val_acc: 0.9760
Epoch 82/100
100/100 [==============================] - 16s 155ms/step - loss: 6.0301e-04 - ac
c: 0.9995 - val_loss: 0.2496 - val_acc: 0.9770
Epoch 83/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0127 - acc: 0.
9980 - val_loss: 0.2524 - val_acc: 0.9760
Epoch 84/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0125 - acc: 0.
9975 - val_loss: 0.2514 - val_acc: 0.9760
Epoch 85/100
100/100 [==============================] - 15s 154ms/step - loss: 0.0083 - acc: 0.
9985 - val_loss: 0.2289 - val_acc: 0.9740
Epoch 86/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0084 - acc: 0.
9985 - val_loss: 0.2519 - val_acc: 0.9730
Epoch 87/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0053 - acc: 0.
9990 - val_loss: 0.2196 - val_acc: 0.9760
Epoch 88/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0092 - acc: 0.
9970 - val_loss: 0.2424 - val_acc: 0.9710
Epoch 89/100
100/100 [==============================] - 16s 156ms/step - loss: 1.6965e-04 - ac
c: 1.0000 - val_loss: 0.2091 - val_acc: 0.9780
Epoch 90/100
100/100 [==============================] - 15s 155ms/step - loss: 0.0126 - acc: 0.
9970 - val_loss: 0.3038 - val_acc: 0.9650
Epoch 91/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0063 - acc: 0.
9995 - val_loss: 0.2270 - val_acc: 0.9790
Epoch 92/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0053 - acc: 0.
9990 - val_loss: 0.2551 - val_acc: 0.9710
Epoch 93/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0011 - acc: 0.
9990 - val_loss: 0.2495 - val_acc: 0.9720
Epoch 94/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0026 - acc: 0.
9990 - val_loss: 0.2742 - val_acc: 0.9670
Epoch 95/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0035 - acc: 0.
9985 - val_loss: 0.2405 - val_acc: 0.9740
Epoch 96/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0056 - acc: 0.
9985 - val_loss: 0.2347 - val_acc: 0.9730
Epoch 97/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0061 - acc: 0.
9990 - val_loss: 0.2260 - val_acc: 0.9750
Epoch 98/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0015 - acc: 0.
9995 - val_loss: 0.2412 - val_acc: 0.9740
Epoch 99/100
100/100 [==============================] - 16s 156ms/step - loss: 0.0030 - acc: 0.
9995 - val_loss: 0.2838 - val_acc: 0.9700
Epoch 100/100
100/100 [==============================] - 16s 155ms/step - loss: 0.0028 - acc: 0.
9990 - val_loss: 0.2604 - val_acc: 0.9750
```

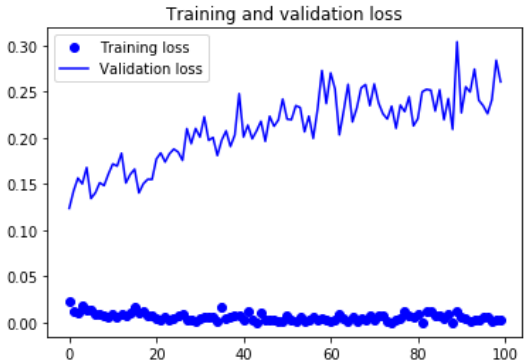
In [16]:

```
model.save('cats_and_dogs_small_4.h5')
```

In [17]:

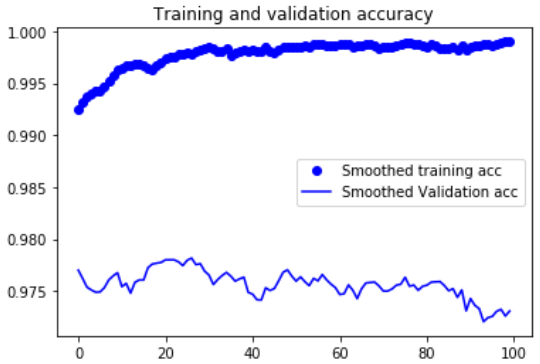
```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```

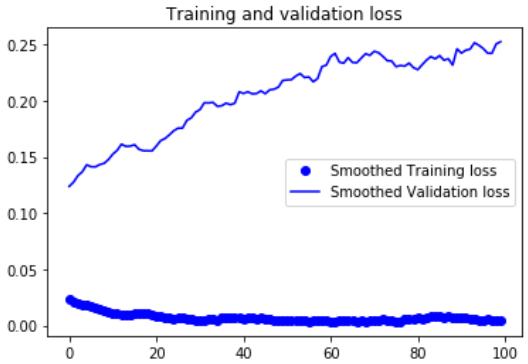




In [19]:

```
def smooth_curve(points, factor=0.8):
    smoothed_points = []
    for point in points:
        if smoothed_points:
            previous = smoothed_points[-1]
            smoothed_points.append(previous * factor + point * (1 - factor))
        else:
            smoothed_points.append(point)
    return smoothed_points

plt.plot(epochs, smooth_curve(acc), 'bo', label='Smoothed training acc')
plt.plot(epochs, smooth_curve(val_acc), 'b', label='Smoothed Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

plt.figure()

plt.plot(epochs, smooth_curve(loss), 'bo', label='Smoothed Training loss')
plt.plot(epochs, smooth_curve(val_loss), 'b', label='Smoothed Validation loss')
plt.title('Training and validation loss')
plt.legend()

plt.show()
```





In [20]:

```
test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=(150,150),
    batch_size=20,
    class_mode='binary')

test_loss, test_acc = model.evaluate_generator(test_generator, steps=50)
print('test acc:', test_acc)
```

```
Found 1000 images belonging to 2 classes.
test acc: 0.9659999918937683
```

In [ ]:

```

```