In [6]:

```
import os, shutil
import re

original_dataset_dir = './train'

base_dir = './datasets/cats_and_dogs_small'
#os.mkdir(base_dir)

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

train_cats_dir = os.path.join(train_dir, 'cats')
os.mkdir(train_cats_dir)

train_dogs_dir = os.path.join(train_dir, 'dogs')
os.mkdir(train_dogs_dir)

validation_cats_dir = os.path.join(validation_dir, 'cats')
os.mkdir(validation_cats_dir)

validation_dogs_dir = os.path.join(validation_dir, 'dogs')
os.mkdir(validation_dogs_dir)

test_cats_dir = os.path.join(test_dir, 'cats')
os.mkdir(test_cats_dir)

test_dogs_dir = os.path.join(test_dir, 'dogs')
os.mkdir(test_dogs_dir)

fnames = ['cat.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_cats_dir, fname)
    shutil.copyfile(src, dst)

fnames = ['cat.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_cats_dir, fname)
    shutil.copyfile(src, dst)

fnames = ['cat.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_cats_dir, fname)
    shutil.copyfile(src, dst)

fnames = ['dog.{}.jpg'.format(i) for i in range(1000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(train_dogs_dir, fname)
    shutil.copyfile(src, dst)

fnames = ['dog.{}.jpg'.format(i) for i in range(1000, 1500)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(validation_dogs_dir, fname)
    shutil.copyfile(src, dst)

fnames = ['dog.{}.jpg'.format(i) for i in range(1500, 2000)]
for fname in fnames:
    src = os.path.join(original_dataset_dir, fname)
    dst = os.path.join(test_dogs_dir, fname)
    shutil.copyfile(src, dst)
```

In [7]:

```
print('훈련용 고양이 이미지 전체 개수:', len(os.listdir(train_cats_dir)))
print('훈련용  강아지 이미지 전체 개수:', len(os.listdir(train_dogs_dir)))
print('검증용 고양이 이미지 전체 개수:', len(os.listdir(validation_cats_dir)))
print('검증용 강아지 이미지 전체 개수:', len(os.listdir(validation_dogs_dir)))
print('테스트용 고양이 이미지 전체 개수:', len(os.listdir(test_cats_dir)))
print('테스트용 강아지 이미지 전체 개수:', len(os.listdir(test_dogs_dir)))
```

```
훈련용 고양이 이미지 전체 개수: 1000
훈련용  강아지 이미지 전체 개수: 1000
검증용 고양이 이미지 전체 개수: 500
검증용 강아지 이미지 전체 개수: 500
테스트용 고양이 이미지 전체 개수: 500
테스트용 강아지 이미지 전체 개수: 500
```

In [13]:

```
from keras import models, layers

model = models.Sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))
```

```
Using TensorFlow backend.

WARNING:tensorflow:From C:WProgramDataWAnaconda3WlibWsite-packagesWtensorflowWpyth
onWframeworkWop_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.
```

In [14]:

```
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 17, 17, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 15, 15, 128)       147584
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 6272)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3211776
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 513
=================================================================
Total params: 3,453,121
Trainable params: 3,453,121
Non-trainable params: 0
_________________________________________________________________
```

In [15]:

```
from keras import optimizers

model.compile(optimizer=optimizers.RMSprop(lr=1e-4),
              loss='binary_crossentropy',
              metrics=['acc'])
```

ImageDataGenerator를 사용하여 디렉터리에서 이미지 읽기

In [16]:

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale=1./255)
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), b
atch_size=20, class_mode='binary')
```

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

In [17]:

```
for data_batch, labels_batch in train_generator:
    print('배치 데이터 크기:', data_batch.shape)
    print('배치 레이블 크기:', labels_batch.shape)
    break
```

```
배치 데이터 크기: (20, 150, 150, 3)
배치 레이블 크기: (20,)
```

In [18]:

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

```
WARNING:tensorflow:From C:WProgramDataWAnaconda3WlibWsite-packagesWtensorflowWpyth
onWopsWmath_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.
Epoch 1/30
100/100 [==============================] - 11s 105ms/step - loss: 0.6896 - acc: 0.
5390 - val_loss: 0.7076 - val_acc: 0.5000
Epoch 2/30
100/100 [==============================] - 7s 75ms/step - loss: 0.6616 - acc: 0.60
85 - val_loss: 0.6342 - val_acc: 0.6370
Epoch 3/30
100/100 [==============================] - 7s 72ms/step - loss: 0.6151 - acc: 0.65
80 - val_loss: 0.6278 - val_acc: 0.6330
Epoch 4/30
100/100 [==============================] - 7s 71ms/step - loss: 0.5655 - acc: 0.71
35 - val_loss: 0.5919 - val_acc: 0.6750
Epoch 5/30
100/100 [==============================] - 7s 71ms/step - loss: 0.5318 - acc: 0.73
80 - val_loss: 0.6352 - val_acc: 0.6690
Epoch 6/30
100/100 [==============================] - 7s 71ms/step - loss: 0.5093 - acc: 0.75
10 - val_loss: 0.6182 - val_acc: 0.6610
Epoch 7/30
100/100 [==============================] - 7s 71ms/step - loss: 0.4772 - acc: 0.77
65 - val_loss: 0.5466 - val_acc: 0.7180
Epoch 8/30
100/100 [==============================] - 7s 71ms/step - loss: 0.4520 - acc: 0.78
40 - val_loss: 0.5375 - val_acc: 0.7300
Epoch 9/30
100/100 [==============================] - 7s 71ms/step - loss: 0.4235 - acc: 0.80
40 - val_loss: 0.5227 - val_acc: 0.7420
Epoch 10/30
100/100 [==============================] - 7s 71ms/step - loss: 0.4041 - acc: 0.81
45 - val_loss: 0.5484 - val_acc: 0.7170
Epoch 11/30
100/100 [==============================] - 7s 71ms/step - loss: 0.3887 - acc: 0.81
85 - val_loss: 0.5234 - val_acc: 0.7360
Epoch 12/30
100/100 [==============================] - 7s 70ms/step - loss: 0.3662 - acc: 0.83
75 - val_loss: 0.5244 - val_acc: 0.7410
Epoch 13/30
100/100 [==============================] - 7s 71ms/step - loss: 0.3335 - acc: 0.85
65 - val_loss: 0.5665 - val_acc: 0.7330
Epoch 14/30
100/100 [==============================] - 7s 71ms/step - loss: 0.3152 - acc: 0.86
50 - val_loss: 0.5561 - val_acc: 0.7400
Epoch 15/30
100/100 [==============================] - 7s 71ms/step - loss: 0.2886 - acc: 0.88
20 - val_loss: 0.5579 - val_acc: 0.7390
Epoch 16/30
100/100 [==============================] - 7s 72ms/step - loss: 0.2640 - acc: 0.88
85 - val_loss: 0.5723 - val_acc: 0.7510
Epoch 17/30
100/100 [==============================] - 7s 72ms/step - loss: 0.2503 - acc: 0.89
95 - val_loss: 0.6238 - val_acc: 0.7360
Epoch 18/30
100/100 [==============================] - 7s 71ms/step - loss: 0.2317 - acc: 0.90
75 - val_loss: 0.6563 - val_acc: 0.7250
Epoch 19/30
100/100 [==============================] - 7s 71ms/step - loss: 0.2137 - acc: 0.91
```

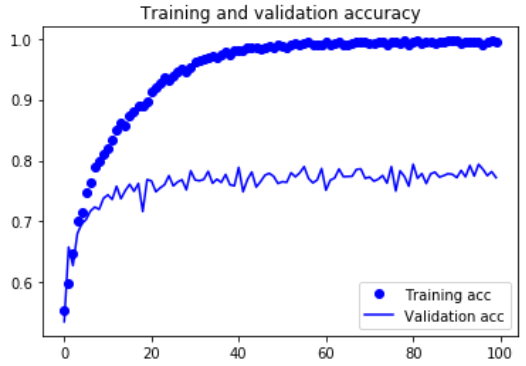
```
75 - val_loss: 0.5955 - val_acc: 0.7330
Epoch 20/30
100/100 [==============================] - 7s 71ms/step - loss: 0.1906 - acc: 0.92
75 - val_loss: 0.6033 - val_acc: 0.7560
Epoch 21/30
100/100 [==============================] - 7s 71ms/step - loss: 0.1697 - acc: 0.93
70 - val_loss: 0.6546 - val_acc: 0.7290
Epoch 22/30
100/100 [==============================] - 7s 71ms/step - loss: 0.1539 - acc: 0.94
90 - val_loss: 0.7131 - val_acc: 0.7420
Epoch 23/30
100/100 [==============================] - 7s 71ms/step - loss: 0.1374 - acc: 0.94
95 - val_loss: 0.6946 - val_acc: 0.7480
Epoch 24/30
100/100 [==============================] - 7s 72ms/step - loss: 0.1194 - acc: 0.95
90 - val_loss: 0.7112 - val_acc: 0.7510
Epoch 25/30
100/100 [==============================] - 7s 71ms/step - loss: 0.1092 - acc: 0.96
65 - val_loss: 0.7151 - val_acc: 0.7500
Epoch 26/30
100/100 [==============================] - 7s 71ms/step - loss: 0.0945 - acc: 0.97
30 - val_loss: 0.7643 - val_acc: 0.7370
Epoch 27/30
100/100 [==============================] - 7s 71ms/step - loss: 0.0816 - acc: 0.97
80 - val_loss: 0.7827 - val_acc: 0.7500
Epoch 28/30
100/100 [==============================] - 7s 71ms/step - loss: 0.0672 - acc: 0.98
00 - val_loss: 0.8500 - val_acc: 0.7390
Epoch 29/30
100/100 [==============================] - 7s 70ms/step - loss: 0.0596 - acc: 0.98
45 - val_loss: 0.8863 - val_acc: 0.7470
Epoch 30/30
100/100 [==============================] - 7s 71ms/step - loss: 0.0530 - acc: 0.98
55 - val_loss: 0.9541 - val_acc: 0.7340
```

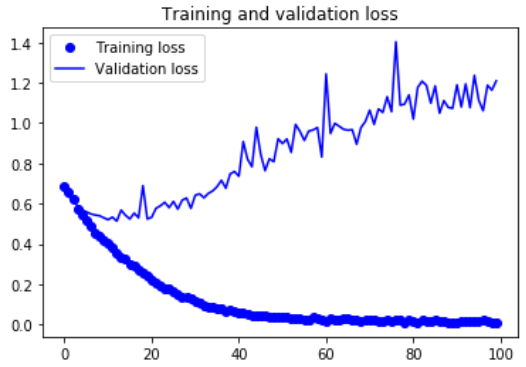
In [20]:

```
model.save('cats_and_dogs_small_1.h5')
```

In [30]:

```
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))

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()
```



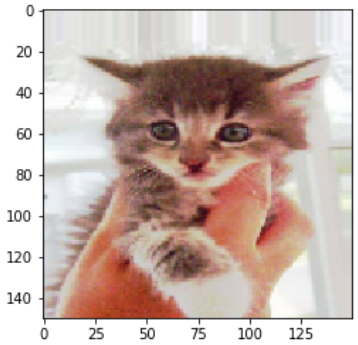



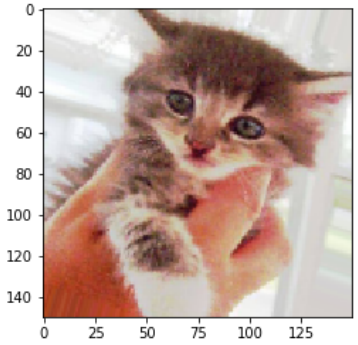
In [23]:

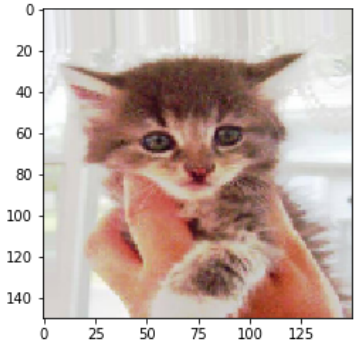
```
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

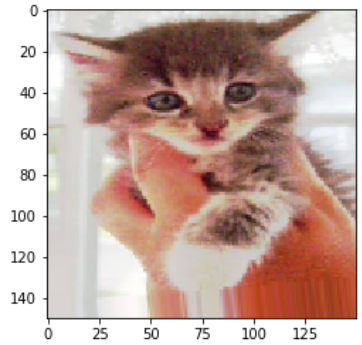
In [24]:

```
from keras.preprocessing import image

fnames = sorted([os.path.join(train_cats_dir, fname) for fname in os.listdir(train_cats_dir)])

img_path = fnames[3]
img = image.load_img(img_path, target_size=(150,150))

x = image.img_to_array(img)
x = x.reshape((1,) + x.shape)

i = 0
for batch in datagen.flow(x, batch_size=1):
    plt.figure(i)
    imgplot = plt.imshow(image.array_to_img(batch[0]))
    i += 1
    if i % 4 == 0:
        break

plt.show()
```









In [25]:

```
model = models.Sequential()
model.add(layers.Conv2D(32, (3,3), activation='relu', input_shape=(150,150,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(128, (3,3), activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer=optimizers.RMSprop(lr=1e-4),
              loss='binary_crossentropy',
              metrics=['acc'])
```

```
WARNING:tensorflow:From C:WProgramDataWAnaconda3WlibWsite-packagesWkerasWbackendWt
ensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) wit
h keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_p
rob`.
```

In [33]:

```
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,)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=(150,150),
    batch_size=32,
    class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(150,150),
    batch_size=32,
    class_mode='binary')

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

```
Found 2000 images belonging to 2 classes.
Found 1000 images belonging to 2 classes.
Epoch 1/100
100/100 [==============================] - 27s 268ms/step - loss: 0.8291 - acc: 0.
6797 - val_loss: 0.5424 - val_acc: 0.7176
Epoch 2/100
100/100 [==============================] - 24s 243ms/step - loss: 0.5961 - acc: 0.
7041 - val_loss: 0.5354 - val_acc: 0.7384
Epoch 3/100
100/100 [==============================] - 24s 243ms/step - loss: 0.5732 - acc: 0.
7113 - val_loss: 0.4741 - val_acc: 0.7741
Epoch 4/100
100/100 [==============================] - 24s 241ms/step - loss: 0.5377 - acc: 0.
7375 - val_loss: 0.4759 - val_acc: 0.7784
Epoch 5/100
100/100 [==============================] - 24s 245ms/step - loss: 0.5311 - acc: 0.
7369 - val_loss: 0.4718 - val_acc: 0.7697
Epoch 6/100
100/100 [==============================] - 25s 246ms/step - loss: 0.5210 - acc: 0.
7444 - val_loss: 0.4492 - val_acc: 0.7854
Epoch 7/100
100/100 [==============================] - 24s 242ms/step - loss: 0.5123 - acc: 0.
7450 - val_loss: 0.4856 - val_acc: 0.7627
Epoch 8/100
100/100 [==============================] - 24s 244ms/step - loss: 0.5093 - acc: 0.
7556 - val_loss: 0.4636 - val_acc: 0.7893
Epoch 9/100
100/100 [==============================] - 24s 243ms/step - loss: 0.5138 - acc: 0.
7491 - val_loss: 0.4397 - val_acc: 0.8061
Epoch 10/100
100/100 [==============================] - 24s 244ms/step - loss: 0.5018 - acc: 0.
7553 - val_loss: 0.4699 - val_acc: 0.7912
Epoch 11/100
100/100 [==============================] - 24s 243ms/step - loss: 0.4825 - acc: 0.
7744 - val_loss: 0.4496 - val_acc: 0.7919
Epoch 12/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4904 - acc: 0.
7700 - val_loss: 0.4231 - val_acc: 0.8058
Epoch 13/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4920 - acc: 0.
7550 - val_loss: 0.5046 - val_acc: 0.7687
Epoch 14/100
100/100 [==============================] - 24s 239ms/step - loss: 0.4740 - acc: 0.
7735 - val_loss: 0.4167 - val_acc: 0.8077
Epoch 15/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4813 - acc: 0.
7694 - val_loss: 0.4255 - val_acc: 0.8112
Epoch 16/100
100/100 [==============================] - 24s 239ms/step - loss: 0.4783 - acc: 0.
7700 - val_loss: 0.4173 - val_acc: 0.8280
Epoch 17/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4703 - acc: 0.
7766 - val_loss: 0.4416 - val_acc: 0.7976
Epoch 18/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4902 - acc: 0.
7672 - val_loss: 0.4744 - val_acc: 0.7745
Epoch 19/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4624 - acc: 0.
7769 - val_loss: 0.4928 - val_acc: 0.7786
Epoch 20/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4625 - acc: 0.
```

```
7809 - val_loss: 0.4095 - val_acc: 0.8402
Epoch 21/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4653 - acc: 0.
7719 - val_loss: 0.4217 - val_acc: 0.8096
Epoch 22/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4666 - acc: 0.
7781 - val_loss: 0.4263 - val_acc: 0.8041
Epoch 23/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4512 - acc: 0.
7831 - val_loss: 0.4141 - val_acc: 0.8179
Epoch 24/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4476 - acc: 0.
7953 - val_loss: 0.4069 - val_acc: 0.8325
Epoch 25/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4446 - acc: 0.
7878 - val_loss: 0.5022 - val_acc: 0.7693
Epoch 26/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4426 - acc: 0.
7944 - val_loss: 0.4124 - val_acc: 0.8166
Epoch 27/100
100/100 [==============================] - 24s 239ms/step - loss: 0.4585 - acc: 0.
7734 - val_loss: 0.4067 - val_acc: 0.8215
Epoch 28/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4467 - acc: 0.
7903 - val_loss: 0.4012 - val_acc: 0.8445
Epoch 29/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4448 - acc: 0.
7972 - val_loss: 0.4490 - val_acc: 0.8048
Epoch 30/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4496 - acc: 0.
7947 - val_loss: 0.4198 - val_acc: 0.8217
Epoch 31/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4341 - acc: 0.
7963 - val_loss: 0.4443 - val_acc: 0.7977
Epoch 32/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4370 - acc: 0.
7994 - val_loss: 0.5497 - val_acc: 0.7558
Epoch 33/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4245 - acc: 0.
8019 - val_loss: 0.4939 - val_acc: 0.8065
Epoch 34/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4394 - acc: 0.
7897 - val_loss: 0.4531 - val_acc: 0.8119
Epoch 35/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4329 - acc: 0.
8009 - val_loss: 0.4035 - val_acc: 0.8103
Epoch 36/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4288 - acc: 0.
8025 - val_loss: 0.3872 - val_acc: 0.8170
Epoch 37/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4216 - acc: 0.
8041 - val_loss: 0.3943 - val_acc: 0.8325
Epoch 38/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4383 - acc: 0.
7916 - val_loss: 0.4945 - val_acc: 0.7854
Epoch 39/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4245 - acc: 0.
8013 - val_loss: 0.4214 - val_acc: 0.8154
Epoch 40/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4159 - acc: 0.
8075 - val_loss: 0.3936 - val_acc: 0.8209
Epoch 41/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4093 - acc: 0.
8219 - val_loss: 0.4461 - val_acc: 0.8099
Epoch 42/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4242 - acc: 0.
8059 - val_loss: 0.3703 - val_acc: 0.8452
Epoch 43/100
100/100 [==============================] - 24s 239ms/step - loss: 0.4126 - acc: 0.
8169 - val_loss: 0.4034 - val_acc: 0.8260
Epoch 44/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4070 - acc: 0.
8159 - val_loss: 0.4342 - val_acc: 0.8027
Epoch 45/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4191 - acc: 0.
8009 - val_loss: 0.4132 - val_acc: 0.8318
Epoch 46/100
100/100 [==============================] - 24s 241ms/step - loss: 0.4139 - acc: 0.
8122 - val_loss: 0.4012 - val_acc: 0.8192
Epoch 47/100
100/100 [==============================] - 24s 242ms/step - loss: 0.4068 - acc: 0.
8144 - val_loss: 0.3974 - val_acc: 0.8183
Epoch 48/100
100/100 [==============================] - 24s 239ms/step - loss: 0.4094 - acc: 0.
8137 - val_loss: 0.4082 - val_acc: 0.8138
Epoch 49/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3886 - acc: 0.
8241 - val_loss: 0.3852 - val_acc: 0.8306
Epoch 50/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3936 - acc: 0.
8184 - val_loss: 0.4672 - val_acc: 0.7822
Epoch 51/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3881 - acc: 0.
8297 - val_loss: 0.3749 - val_acc: 0.8325
Epoch 52/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3933 - acc: 0.
8178 - val_loss: 0.4152 - val_acc: 0.8189
Epoch 53/100
100/100 [==============================] - 24s 240ms/step - loss: 0.4035 - acc: 0.
8131 - val_loss: 0.3693 - val_acc: 0.8395
Epoch 54/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3950 - acc: 0.
8216 - val_loss: 0.5270 - val_acc: 0.7713
Epoch 55/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3958 - acc: 0.
8259 - val_loss: 0.3972 - val_acc: 0.8261
Epoch 56/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3933 - acc: 0.
8231 - val_loss: 0.3983 - val_acc: 0.8305
Epoch 57/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3896 - acc: 0.
8231 - val_loss: 0.3751 - val_acc: 0.8351
Epoch 58/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3810 - acc: 0.
8231 - val_loss: 0.3620 - val_acc: 0.8458
Epoch 59/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3677 - acc: 0.
8406 - val_loss: 0.3979 - val_acc: 0.8383
Epoch 60/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3957 - acc: 0.
8203 - val_loss: 0.3722 - val_acc: 0.8388
Epoch 61/100
```

```
100/100 [==============================] - 24s 241ms/step - loss: 0.3711 - acc: 0.
8316 - val_loss: 0.3626 - val_acc: 0.8492
Epoch 62/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3877 - acc: 0.
8275 - val_loss: 0.3824 - val_acc: 0.8471
Epoch 63/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3789 - acc: 0.
8266 - val_loss: 0.5055 - val_acc: 0.7880
Epoch 64/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3665 - acc: 0.
8366 - val_loss: 0.3556 - val_acc: 0.8537
Epoch 65/100
100/100 [==============================] - 24s 238ms/step - loss: 0.3800 - acc: 0.
8338 - val_loss: 0.4169 - val_acc: 0.8179
Epoch 66/100
100/100 [==============================] - 24s 243ms/step - loss: 0.3697 - acc: 0.
8372 - val_loss: 0.3910 - val_acc: 0.8164
Epoch 67/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3588 - acc: 0.
8425 - val_loss: 0.4567 - val_acc: 0.8179
Epoch 68/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3638 - acc: 0.
8403 - val_loss: 0.5856 - val_acc: 0.7610
Epoch 69/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3786 - acc: 0.
8285 - val_loss: 0.3723 - val_acc: 0.8452
Epoch 70/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3687 - acc: 0.
8300 - val_loss: 0.4174 - val_acc: 0.8202
Epoch 71/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3637 - acc: 0.
8381 - val_loss: 0.3741 - val_acc: 0.8395
Epoch 72/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3412 - acc: 0.
8516 - val_loss: 0.3972 - val_acc: 0.8305
Epoch 73/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3405 - acc: 0.
8478 - val_loss: 0.5241 - val_acc: 0.8028
Epoch 74/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3615 - acc: 0.
8381 - val_loss: 0.3870 - val_acc: 0.8325
Epoch 75/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3521 - acc: 0.
8478 - val_loss: 0.4539 - val_acc: 0.8086
Epoch 76/100
100/100 [==============================] - 24s 242ms/step - loss: 0.3399 - acc: 0.
8531 - val_loss: 0.3480 - val_acc: 0.8534
Epoch 77/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3554 - acc: 0.
8416 - val_loss: 0.4301 - val_acc: 0.8183
Epoch 78/100
100/100 [==============================] - 24s 242ms/step - loss: 0.3466 - acc: 0.
8491 - val_loss: 0.3613 - val_acc: 0.8426
Epoch 79/100
100/100 [==============================] - 24s 239ms/step - loss: 0.3503 - acc: 0.
8459 - val_loss: 0.3636 - val_acc: 0.8499
Epoch 80/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3571 - acc: 0.
8431 - val_loss: 0.3653 - val_acc: 0.8447
Epoch 81/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3520 - acc: 0.
8456 - val_loss: 0.3637 - val_acc: 0.8357
Epoch 82/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3386 - acc: 0.
8506 - val_loss: 0.4579 - val_acc: 0.8080
Epoch 83/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3422 - acc: 0.
8466 - val_loss: 0.3799 - val_acc: 0.8287
Epoch 84/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3277 - acc: 0.
8628 - val_loss: 0.3936 - val_acc: 0.8447
Epoch 85/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3465 - acc: 0.
8469 - val_loss: 0.4020 - val_acc: 0.8401
Epoch 86/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3326 - acc: 0.
8556 - val_loss: 0.3771 - val_acc: 0.8415
Epoch 87/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3340 - acc: 0.
8578 - val_loss: 0.3869 - val_acc: 0.8350
Epoch 88/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3254 - acc: 0.
8619 - val_loss: 0.4401 - val_acc: 0.8228
Epoch 89/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3276 - acc: 0.
8547 - val_loss: 0.3945 - val_acc: 0.8351
Epoch 90/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3242 - acc: 0.
8569 - val_loss: 0.3974 - val_acc: 0.8522
Epoch 91/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3251 - acc: 0.
8597 - val_loss: 0.3871 - val_acc: 0.8454
Epoch 92/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3151 - acc: 0.
8609 - val_loss: 0.3952 - val_acc: 0.8236
Epoch 93/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3132 - acc: 0.
8625 - val_loss: 0.5019 - val_acc: 0.7784
Epoch 94/100
100/100 [==============================] - 24s 240ms/step - loss: 0.3180 - acc: 0.
8631 - val_loss: 0.4297 - val_acc: 0.8268
Epoch 95/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3293 - acc: 0.
8600 - val_loss: 0.3314 - val_acc: 0.8595
Epoch 96/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3118 - acc: 0.
8694 - val_loss: 0.3483 - val_acc: 0.8537
Epoch 97/100
100/100 [==============================] - 24s 242ms/step - loss: 0.3240 - acc: 0.
8569 - val_loss: 0.4026 - val_acc: 0.8484
Epoch 98/100
100/100 [==============================] - 24s 242ms/step - loss: 0.3325 - acc: 0.
8575 - val_loss: 0.3895 - val_acc: 0.8383
Epoch 99/100
100/100 [==============================] - 24s 241ms/step - loss: 0.3213 - acc: 0.
8666 - val_loss: 0.3403 - val_acc: 0.8598
Epoch 100/100
100/100 [==============================] - 24s 242ms/step - loss: 0.3246 - acc: 0.
8600 - val_loss: 0.3544 - val_acc: 0.8550
```

In [34]:

```
model.save('cats_and_dogs_small_2.h5')
```

In [35]:

```
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()
```