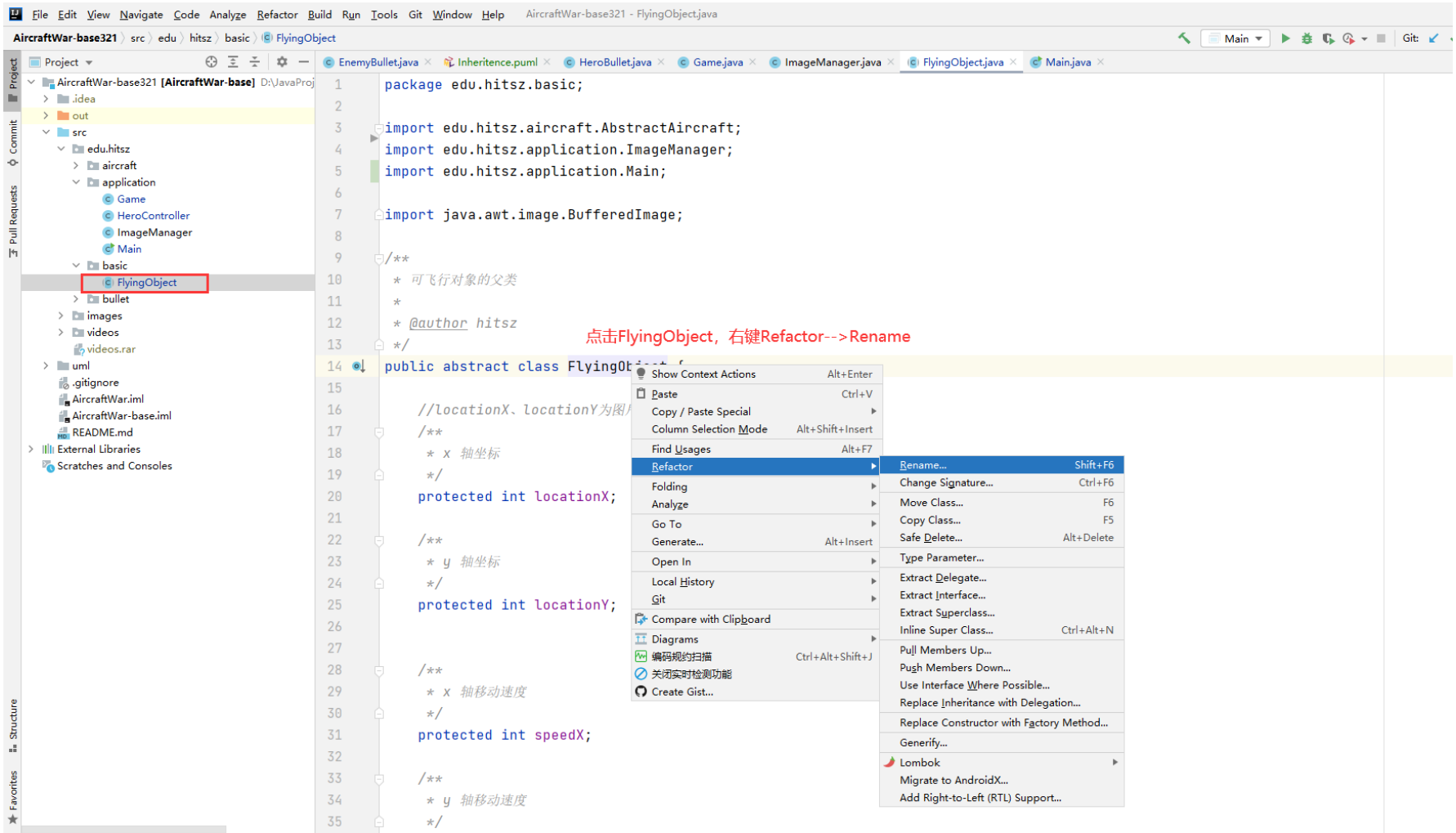
实验二开始前要完成的修改

大家在开始实验二前先完成以下修改：

1、把basic下的FlyingObject改成AbstractFlyingObject；

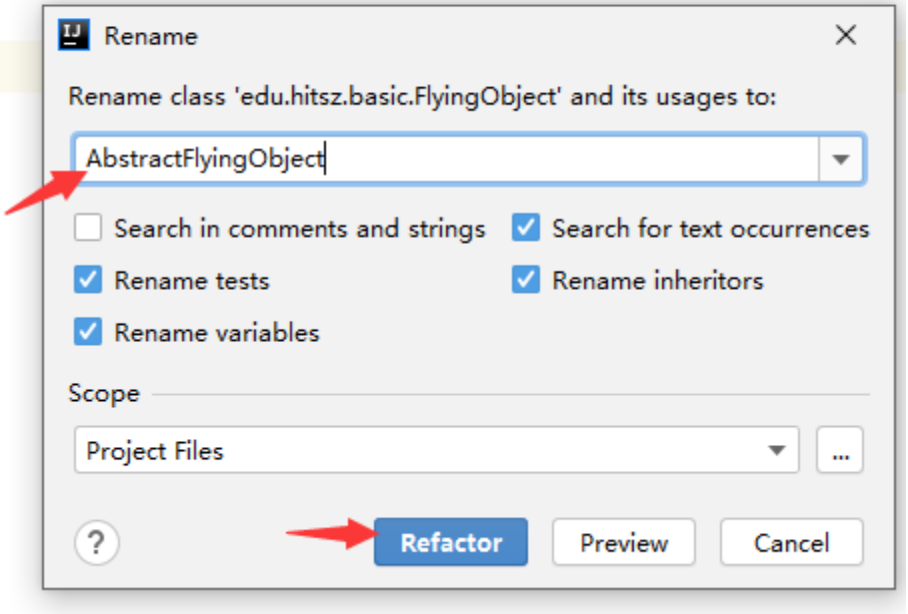


实验二开始前要完成的修改

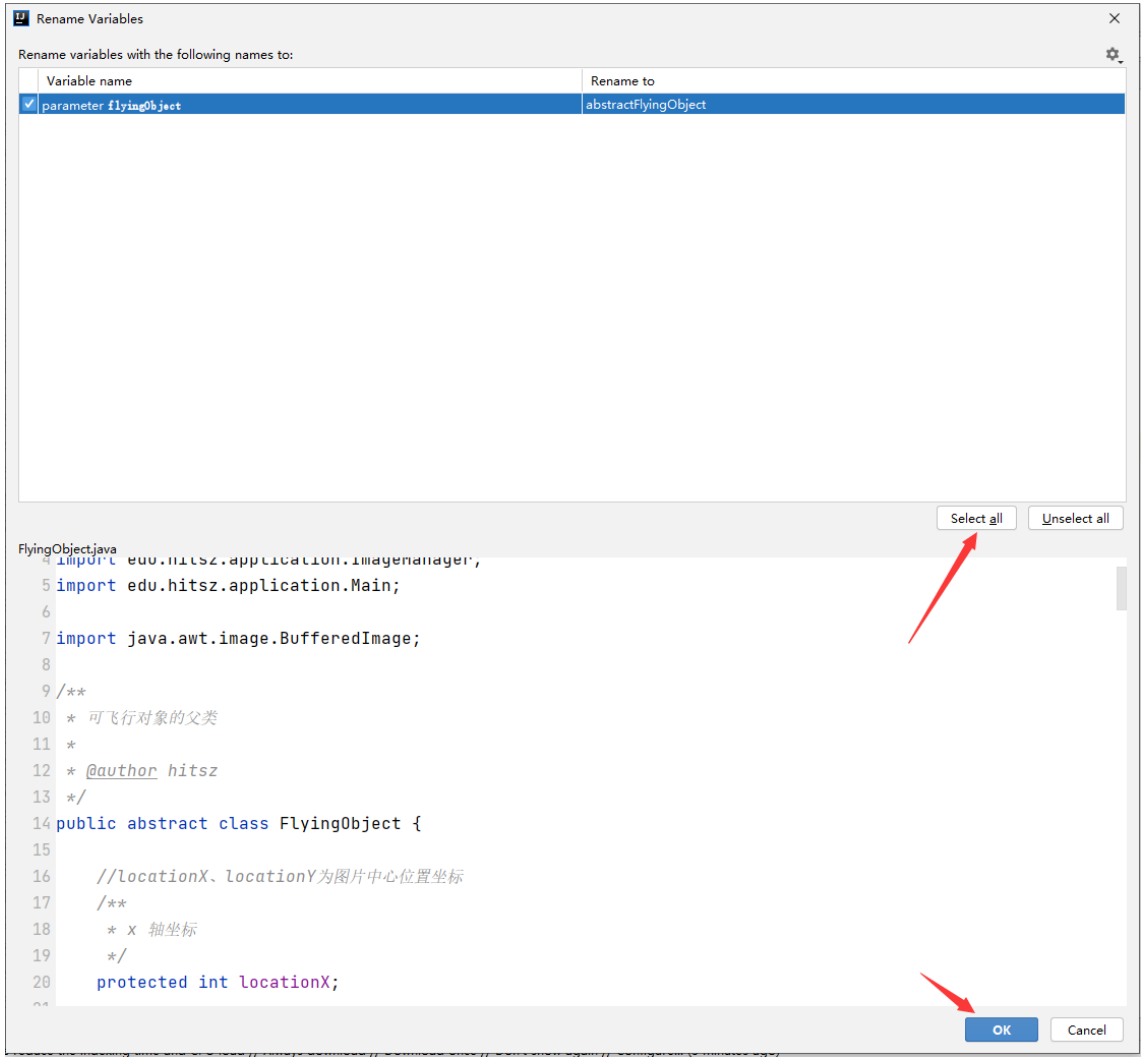
```java
/**
 * 可飞行对象的父类
 *
 * @author hitsz
 */
public abstract class FlyingObject {

    //locationX、locationY为图片中心位置坐标
    /**
     * x 轴坐标
     */
    protected int locationX;

    /**
     * y 轴坐标
     */
```



实验二开始前要完成的修改



实验二开始前要完成的修改

可以看到类名、构造函数、用到这个类的地方都改过来了。

```
 */
public abstract class AbstractFlyingObject {

    //locationX、locationY为图片中心位置坐标
    /**
     * x 轴坐标

     public AbstractFlyingObject() {
     }


 *
 * @param abstractFlyingObject 撞击对方
 * @return true: 我方被击中; false 我方未被击中
 */
public boolean crash(AbstractFlyingObject abstractFlyingObject) {
    // 缩放因子，用于控制 y轴方向区域范围
    int factor = this instanceof AbstractAircraft ? 2 : 1;
```

实验二开始前要完成的修改

大家在开始实验二前先完成以下修改：

1、把basic下的FlyingObject改成AbstractFlyingObject；

2、把bullet下的AbstractBullet 改名为 BaseBullet；（因为这个类不是抽象类，所以叫AbstractBullet不那么合适，严谨一点改成BaseBullet。修改方法跟刚才一模一样，用refactor。）

实验二开始前要完成的修改

可以看到类名、构造函数、用到这个类的地方都改过来了。

```
/**
 * 子弹类。
 * 也可以考虑不同类型的子弹
 *
 * @author hitsz
 */
public class BaseBullet extends AbstractFlyingObject {

    private int power = 10;

    public BaseBullet(int locationX, int locationY, int speedX, int speedY, int power) {
        super(locationX, locationY, speedX, speedY);
        this.power = power;
    }
@Override
/**
 * 通过射击产生子弹
 * @return 射击出的子弹List
 */
public List<BaseBullet> shoot() {
    List<BaseBullet> res = new LinkedList<>();
    int x = this.getLocationX();
    int y = this.getLocationY() + direction*2;
    int speedX = 0;
    int speedY = this.getSpeedY() + direction*5;
    BaseBullet baseBullet;
    for(int i=0; i<shootNum; i++){
        // 子弹发射位置相对飞机位置向前偏移
        // 多个子弹横向分散
        baseBullet = new HeroBullet( locationX: x + (i*2 - shootNum + 1)*10, y, speedX, speedY, power);
```

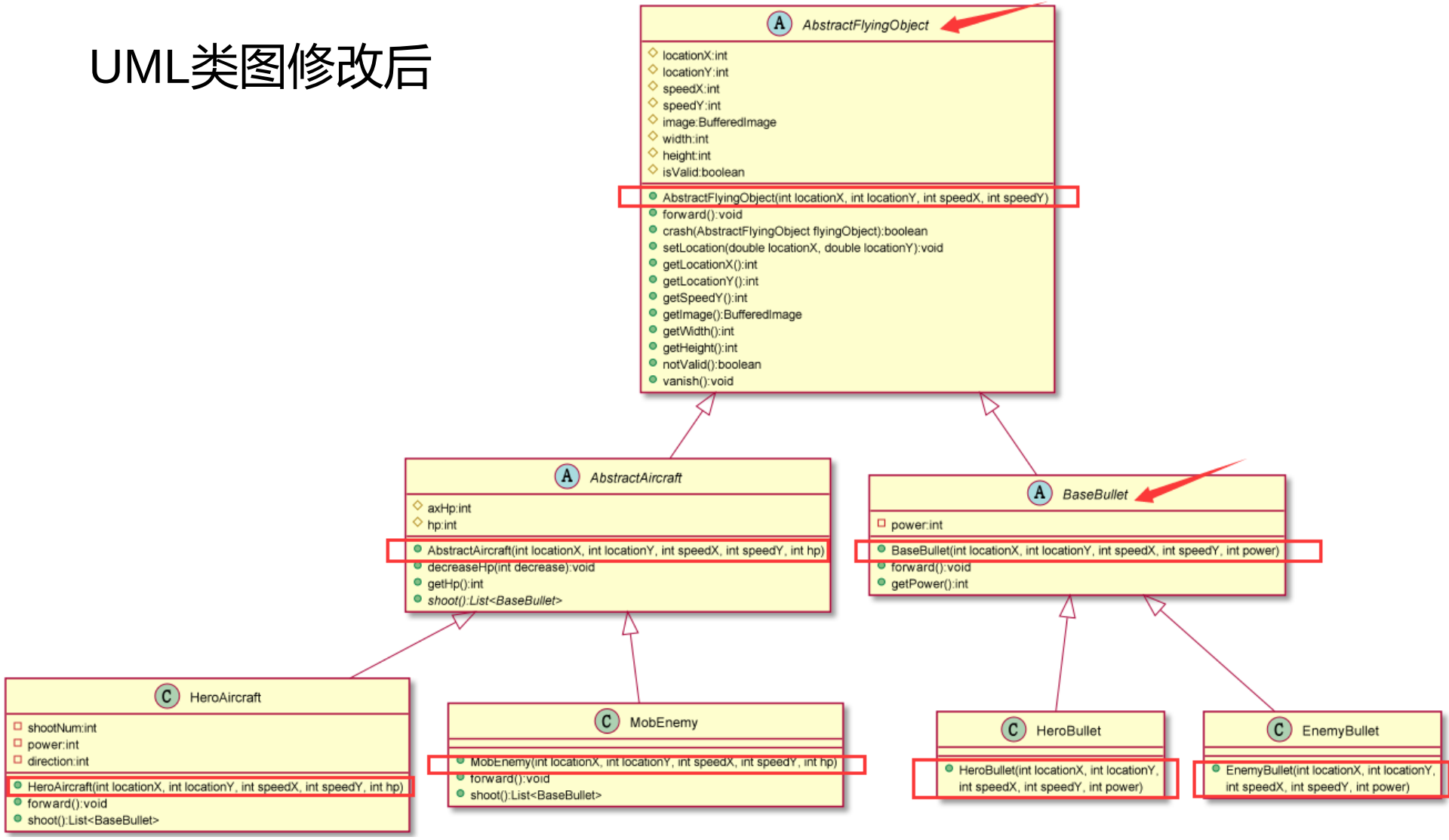
补充说明

大家在开始实验二前先完成以下修改：

1、把basic下的FlyingObject改成AbstractFlyingObject；

2、把bullet下的AbstractBullet 改名为 BaseBullet；

3、按照刚才的命名修改UML类图，并把类图中原来构造函数的返回值都去掉。

补充说明

UML类图修改后