

Lösungen:

```
/**Aufgabe 3**/  
void ViererReiheLegen(){  
    for (int i=1; i<=4; i++){  
        Hinlegen();  
        Schritt();  
    }  
}  
  
/**Aufgabe 4**/  
void ReiheZurWand(){  
    while(IstWand()==false){ //oder while(!IstWand())  
        Hinlegen();  
        Schritt();  
    }  
}  
  
/**Aufgabe 5**/  
void FuenferQuadrat(){  
    for (int i=1; i<=4; i++){  
        ViererReiheLegen();  
        LinksDrehen();  
    }  
}  
  
/**Aufgabe 6**/  
void SchrittNachRechts(){  
    RechtsDrehen();  
    Schritt();  
    LinksDrehen();  
}  
void SchrittNachLinks (){  
    LinksDrehen();  
    Schritt();  
    RechtsDrehen();  
}  
  
/**Aufgabe 7**/  
void MauerBauen(){  
    for (int i=1; i<=5; i++){  
        for (int k=1; k<=4; k++){  
            Hinlegen();  
        }  
        SchrittNachLinks();  
    }  
}
```

```
/**Aufgabe 8**/
void HundertZiegel() {
    InDieEcke()
    for (int i = 1; i <= 100; i++) {
        if (IstWand ()) {
            LinksDrehen();
        }
        Hinlegen();
        Schritt();
    }
}

/**Aufgabe 9**/
void InDieEcke() {
    while (IstBlickNorden() == false) {
        RechtsDrehen();
    }
    for (int i=1; i<=2; i++){
        while (IstWand() == false) {
            Schritt();
        }
        LinksDrehen();
    }
}

/**Aufgabe 10**/
void EbeneLegen() {
    SchrittNachLinks();
    RechtsDrehen();
    Hinlegen();
    Schritt();
    LinksDrehen();
    do {
        while ((!IstWand()) && (!IstZiegel())) {
            Hinlegen();
            Schritt();
        }
        LinksDrehen();
    } while (IstZiegel() == false);
}
```

```
/**Aufgabe 11**/  
class TEST{  
    WELT deutschland;  
    ROBOTER mustermann;  
  
    public TEST(){  
        deutschland = new WELT(15,10,4);  
        mustermann = new ROBOTER(deutschland);  
    }  
  
    void testmethode(){  
        mustermann.FuenferQuadrat();  
        mustermann.ViererReiheLegen();  
    }  
}  
  
/**Aufgabe 12**/  
void ReiheSammeln(){  
    int summe = 0;  
    while (!IstWand()){  
        if(IstZiegel()){  
            Aufheben();  
            summe = summe + 1;  
        }  
        Schritt();  
    }  
    System.out.println(summe);  
}  
  
/**Aufgabe 13**/  
void Zufallsschritt(){  
    int z = (int) (Math.random()*2);  
    if (z==1){  
        Schritt();  
    }  
}  
void wettrennen(){  
    while((a.IstWand()==false)&&(b.IstWand()==false)){  
        int z = (int) (Math.random()*2);  
        if (z==0){  
            a.Zufallsschritt();  
        }else{  
            b.Zufallsschritt();  
        }  
    }  
    if (a.IstWand()){  
        System.out.println("Rob A ist der Sieger");  
        //zaehlera++;  
    }else{  
        System.out.println("Rob B ist der Sieger");  
        //zaehlerb++;  
    }  
}
```

```

/**Aufgabe 14**/
void test20rennen(){
    s=0;
    m=0;
    for (int i=0; i<10;i++){
        wettrennen();
        System.out.println("Neuer Stand: ");
        System.out.println("Stefan: " +s+ " Max: "+m);
        startpos();
    }
}
void startpos(){
    stef.LinksDrehen();
    max.LinksDrehen();
    stef.LinksDrehen();
    max.LinksDrehen();
    while(stef.IstWand()==false && max.IstWand()==false){
        stef.Schritt();
        max.Schritt();
    }
    while(stef.IstWand()==false){
        stef.Schritt();
    }
    while(max.IstWand()==false){
        max.Schritt();
    }
    stef.LinksDrehen();
    max.LinksDrehen();
    stef.LinksDrehen();
    max.LinksDrehen();
}

/**Aufgabe 15**/
void AllesSammeln (){
    while (!IstWand()){
        while (!IstWand ()) {
            ReiheSammeln();
        }
        LinksDrehen();
        if (!IstWand()){
            Schritt ();
            LinksDrehen ();
            while (!IstWand()){
                ReiheSammeln();
            }
            RechtsDrehen();
            if (!IstWand()){
                Schritt ();
                RechtsDrehen ();
            }
        }
    }
}

```

```
/**Aufgabe 16**/  
void wasMachIch(){  
    Schritt();  
    Schritt();  
    for (int i=1; i<=4; i++){  
        while (!IstWand()){  
            LinksDrehen();  
            Hinlegen();  
            RechtsDrehen();  
            Schritt();  
        }  
        LinksDrehen();  
        Schritt();  
        Schritt();  
    }  
}  
  
/**Aufgabe 17**/  
void dreiQuadrate(){  
    for (int i=1; i<=3; i++){  
        for (int k=1; k<=4; k++){  
            Hinlegen();  
            Schritt();  
            LinksDrehen();  
        }  
        Schritt();  
        Schritt();  
        LinksDrehen();  
        Schritt();  
        Schritt();  
        RechtsDrehen();  
    }  
}
```