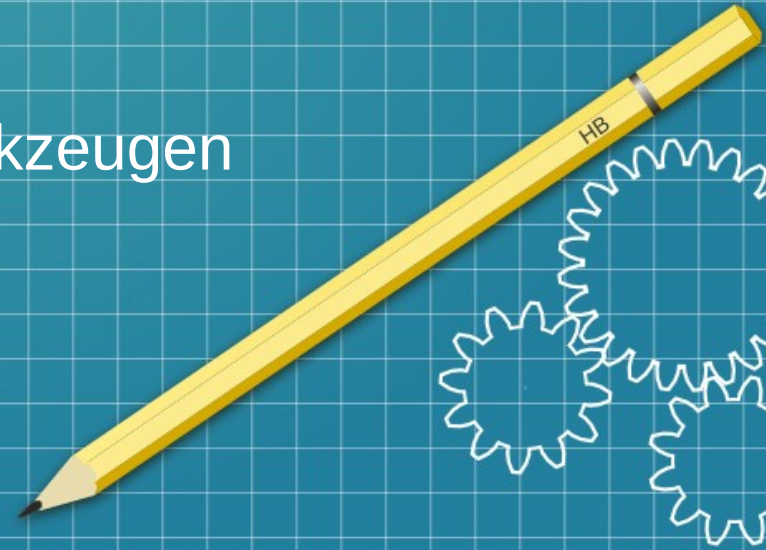
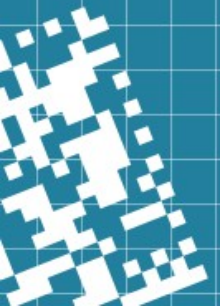
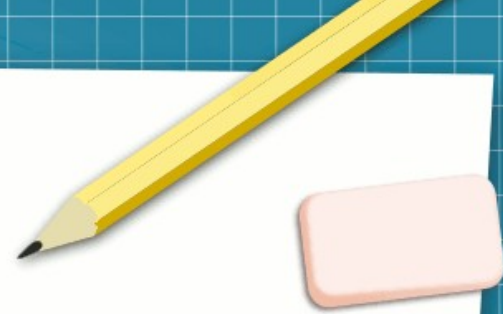


Tools

Einführung von eigenen Werkzeugen

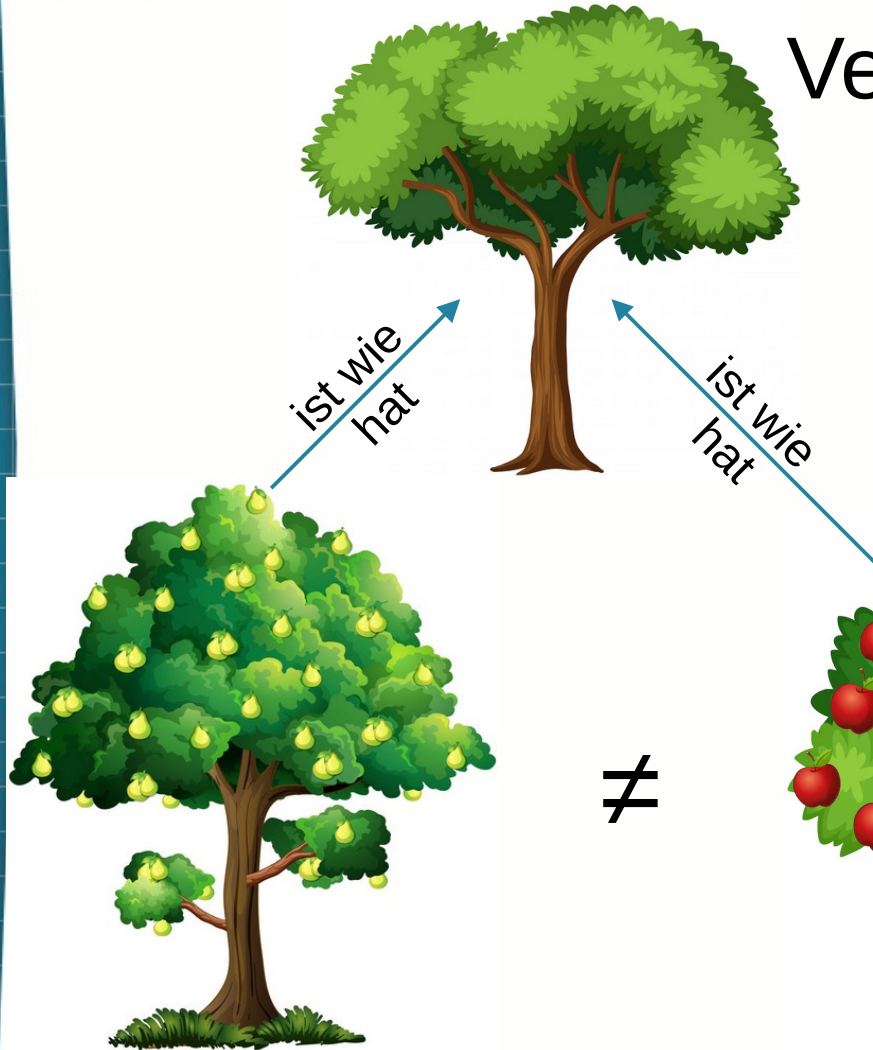


Eigenes Werkzeug

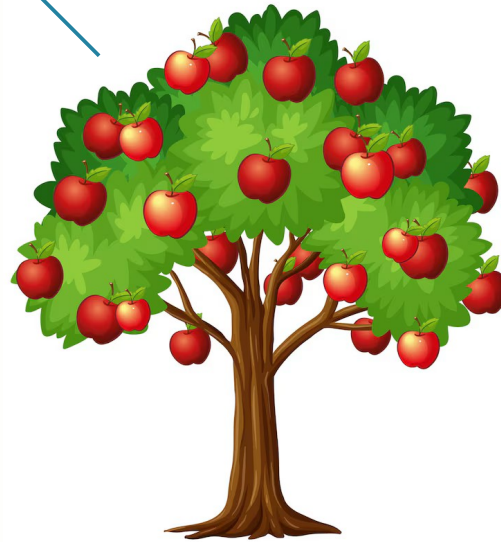


- Neue eigene Item-Klasse, die von Item erbt
 - Klasse anlegen, erweitern, verwenden
- Collections
 - Listen, Maps und Sets
- Loops
 - for, while, foreach
- Idee: der Stummel soll Blöcke zerschlagen
 - Heuballen zu 9 Weizen
 - getrocknete Seetangblöcke in 9 getrockneten Seetang

Vererbung



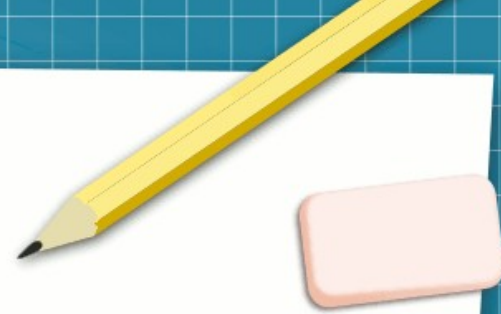
≠



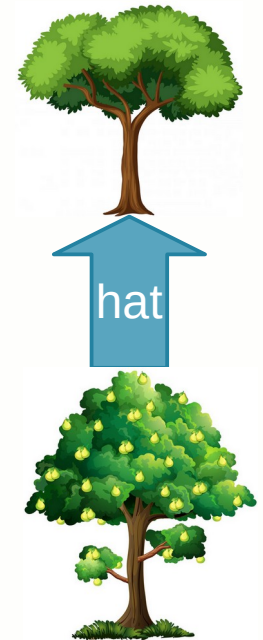
≠



Vererbung bei Klassen

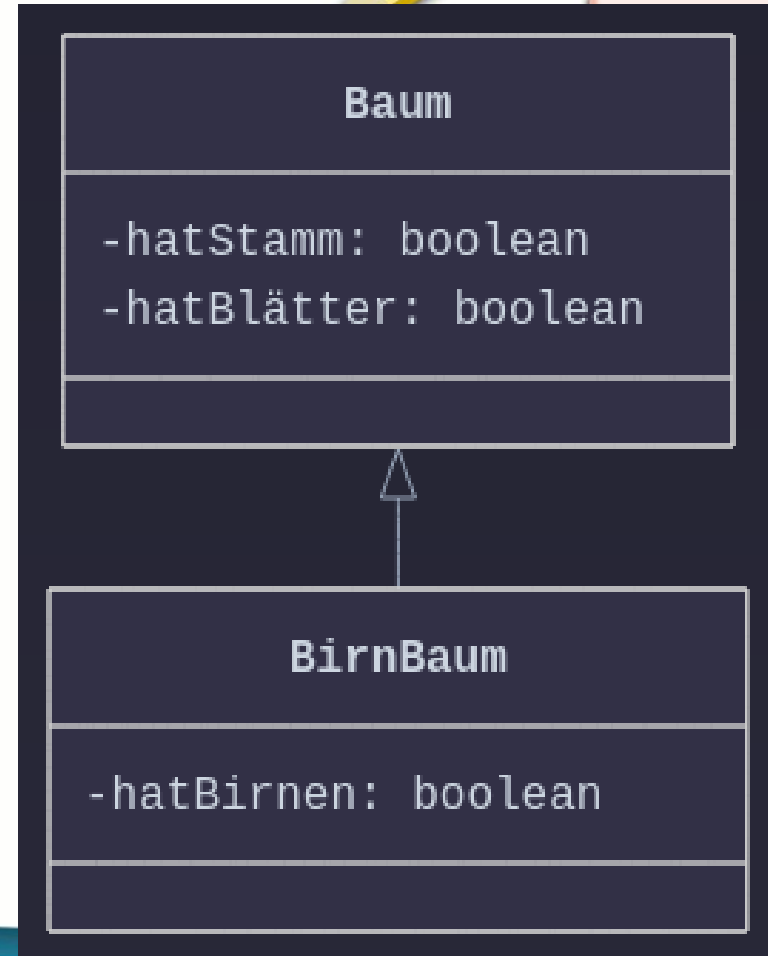


- Klasse Baum
 - Hat einen Stamm
 - lässt vor dem Winter die Blätter fallen
 - Hat Blätter
 - verfärbt sich im Herbst
- Klasse Apfelbaum erbt von Klasse Baum
 - Hat Äpfel
- Klasse Birnbaum erbt von Klasse Baum
 - Hat Birnen
- Klasse Bananenstaude erbt nicht
 - Hat Bananen
 - hat einen Strunk



Beispiel Vererbung im Java-Code

```
public class Baum {  
    private boolean hatStamm = true;  
    private boolean hatBlätter = true;  
}  
  
public class BirnBaum extends Baum {  
    private boolean hatBirnen = true;  
}
```



Stummel-Klasse anlegen



- In `de.jottyfan.minecraft.item`

```
public class StubItem extends Item {  
  
    public StubItem(Settings settings) {  
        super(settings);  
    }  
}
```

Sammlungen (Collections)

- Mehrere gleichartige Elemente wie in einem Korb halten
 - Eier in Eierpackung
 - Limo in Limokiste
- Verschiedene Collection-Typen
 - Listen (gleiche Elemente)
 - Maps (benannte Elemente in einer Liste)
 - Sets (Listen ohne Duplikate)



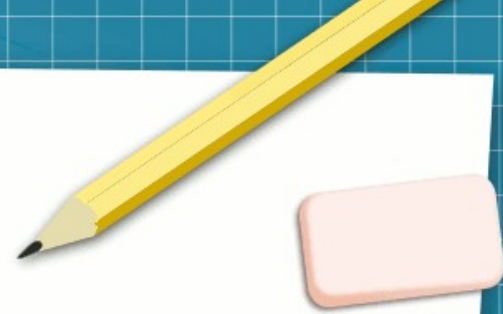
Änderungen in Map registrieren



- Map<Block, Item> als Container für Blöcke, die man mit dem Stummel zerschlagen kann

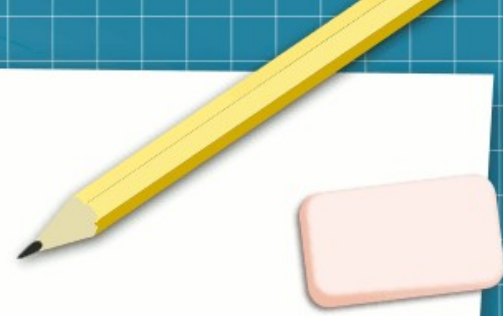
```
private static final Map<Block, Item> SLASH_MAP = Map.of(  
    Blocks.HAY_BLOCK, Items.WHEAT,  
    Blocks.DRIED_KELP_BLOCK, Items.DRIED_KELP);
```


Loops



- Wiederholte Ausführung
 - Zähle rückwärts von 10 bis 1
 - Idee: $10 - 1 = 9$, $9 - 1 = 8$, $8 - 1 = 7$, ... bis $2 - 1 = 1$
 - Als Algorithmus: $X = 10$. $X = X - 1$. Solange machen, wie $X \geq 1$
 - `for (int x = 10; x >= 1; x--) { ausführen }`
 - Alternative: while-Schleife (solange machen, wie gilt: $X \geq 1$)
 - `int x = 10; while (x >= 1) { x--; ausführen }`
 - Collections und Loops
 - Foreach (für alle Elemente in der Liste / dem Set / der Map)
 - `List<Integer> list; for (Integer i : list) { ausführen }`

Programmieridee



- der Stummel soll Blöcke zerschlagen
 - Heuballen zu 9 Weizen
 - getrocknete Seetangblöcke in 9 getrockneten Seetang
- Verwenden der for-Schleife (1 – 9 Items pro Block)
- Verwenden der Map (Ersetzen des richtigen Items pro Block)

Das Zerschlagen programmieren

```
@Override
public ActionResult useOnBlock(ItemUsageContext context) {
    World world = context.getWorld();
    BlockPos pos = context.getBlockPos();
    Block clickedBlock = world.getBlockState(pos).getBlock();
    if (SLASH_MAP.containsKey(clickedBlock)) {
        if (!world.isClient()) {
            world.setBlockState(pos, Blocks.AIR.getDefaultState());
            world.playSound(null, pos, SoundEvents.ITEM_HOE_TILL,
                SoundCategory.BLOCKS);
            for (int i = 0; i < 9; i++) {
                ItemStack stack = new ItemStack(SLASH_MAP.get(clickedBlock));
                float scatter = new Random().nextFloat();
                ItemEntity entity = new ItemEntity(world, pos.getX(),
                    pos.getY(), pos.getZ(), stack, scatter, scatter, 0.2);
                world.spawnEntity(entity);
            }
        }
    }
    return ActionResult.SUCCESS;
}
```

Item-Registrierung anpassen

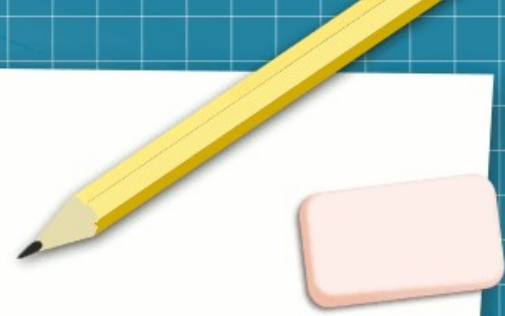


- Registrierung durch eigene Registrierungsmethode ersetzen

```
public static final Item STUB = registerStubItem(  
    Identifier.of(Gtamfmd.MOD_ID, "stub"), new Item.Settings());  
  
private static Item registerStubItem(Identifier identifier, Settings settings) {  
    return Registry.register(Registries.ITEM, identifier, new StubItem(  
        settings.useItemPrefixedTranslationKey()  
        .registryKey(RegistryKey.of(RegistryKeys.ITEM, identifier))));  
}
```

Arbeit sichern

- Git commit und push





This work is licensed under a Creative Commons
Attribution-ShareAlike 3.0 Unported License.
It makes use of the works of Mateus Machado Luna.

