“Tech4Kek”

Lukas Arnold, John Reinecke, Marvin Hof, Thomas Konietzny

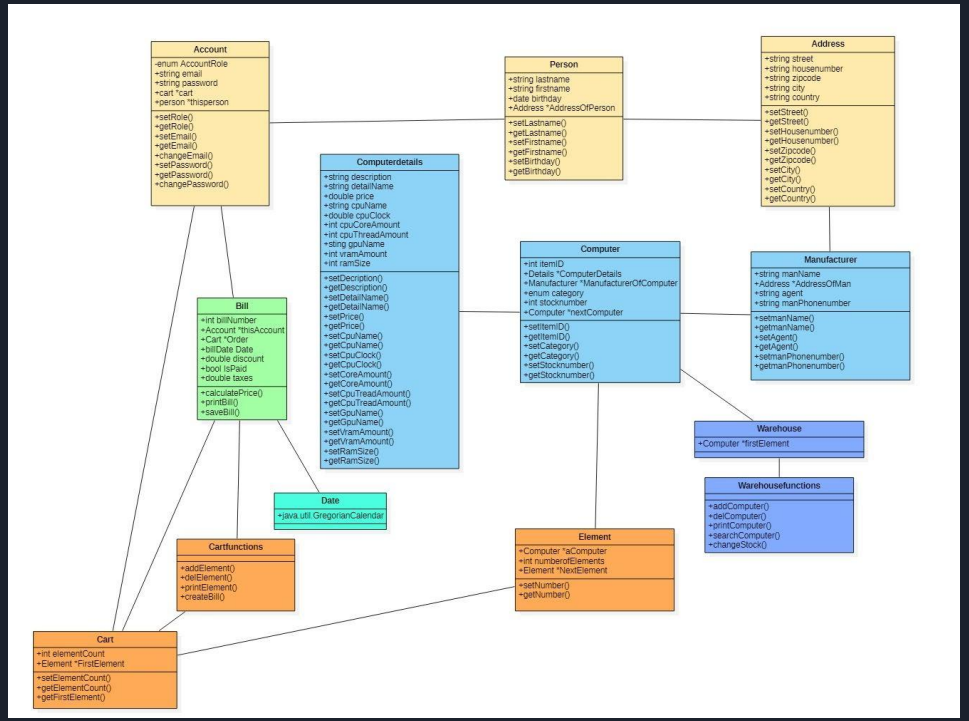
Gliederung

- Unsere Projektidee
- Diagramme
- Codebeispiel
- Unit Tests
- Was folgt?
- Lessons Learned

Unsere Projektidee

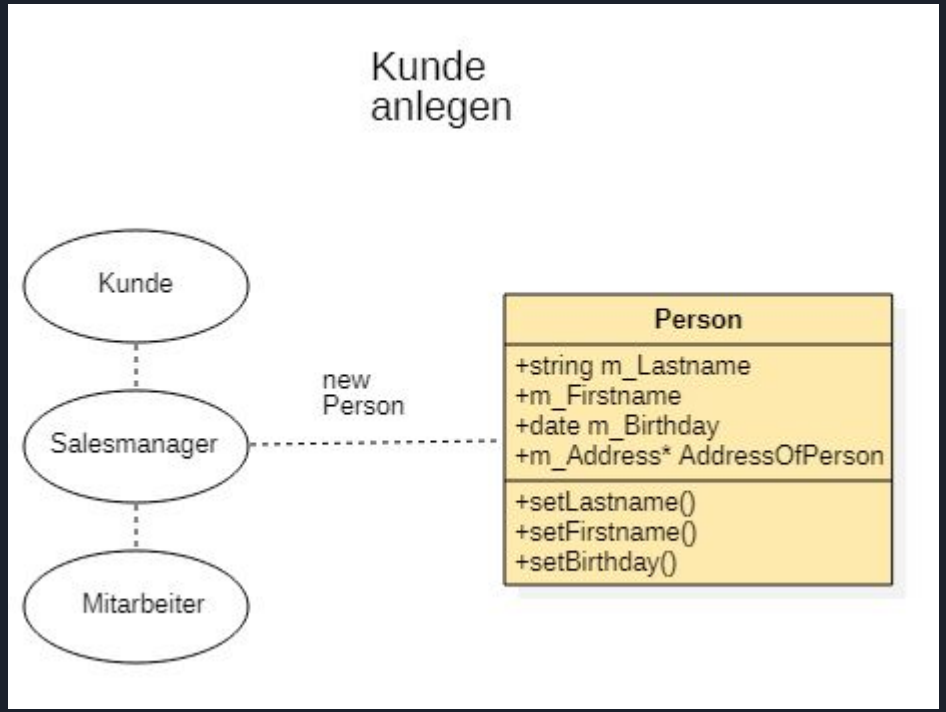
Bei “Tech4Kek” handelt es sich um einen online Handel, welcher sich auf den Verkauf von komplett PCs fokussiert.

Diagramme



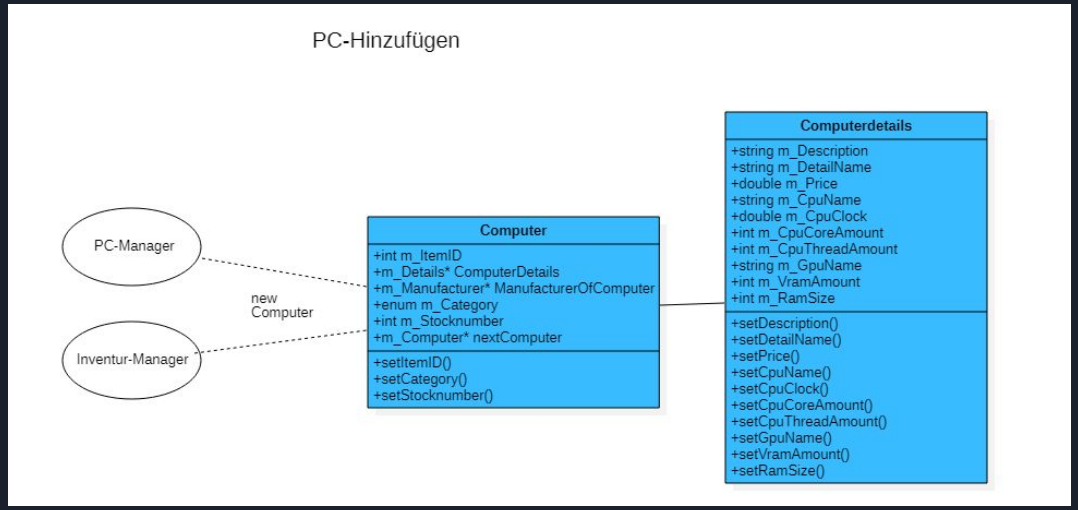
Klassendiagramm

Diagramme



Kunde anlegen

Diagramme



PC-Hinzufügen

Codebeispiel

```java
public class Account {
    private AccountRole m_Role;
    private String m_EMail;
    private String m_Password;
    private Person m_ThePerson;
    private Cart m_TheCart;

    //...

    //Set/Get Functions
    public void SetRole(AccountRole m_Role) { this.m_Role = m_Role; }

    public void SetEMail(String m_EMail) { this.m_EMail = m_EMail; }

    public void SetPassword(String m_Password) { this.m_Password = m_Password; }

    public void SetPerson(Person m_ThePerson) { this.m_ThePerson = m_ThePerson; }

    public AccountRole GetRole() { return m_Role; }

    public String GetEMail() { return m_EMail; }

    public String GetPassword() { return m_Password; }

    public Person GetPerson() { return m_ThePerson; }

    public void setM_TheCart(Cart TheCart) { m_TheCart = TheCart; }

    public Cart getM_TheCart() { return m_TheCart; }

}
```

```java
public double calculate_Price(){
    Element Anchor = m_Order.getM_firstElement();

    while (Anchor.getM_nextElement() != null){
        Anchor = Anchor.getM_nextElement();
        m_Price = m_Price + (Anchor.getM_CountOfComputers() * Anchor.getM_Computer().getM_ComputerDetails().getM_Price());
    }

    double zwierg = 0;
    zwierg = m_Price /100 * m_Discount;
    m_Price = m_Price - zwierg;

    zwierg = m_Price /100 * m_Taxes;
    m_Price = m_Price + zwierg;

    return m_Price;
}
```

```java
public class Warehouse {
    //Constructor
    public Warehouse(){
        firstComputer = new Computer();
        firstComputer.setM_NextComputer(null);
        firstComputer.setM_ManufacturerOfComputer(null);
        firstComputer.setM_ComputerDetails(new Computerdetails());
        firstComputer.getM_ComputerDetails().setM_Description("Anchor");
    }
    //Variable
    Computer firstComputer;

    //Set/Get
    public void setFirstComputer(Computer firstComputer) { this.firstComputer = firstComputer; }

    public Computer getFirstComputer() { return firstComputer; }
}
```

Unit Tests

```
class BillTest {

    @Test
    void calculate_Price() {

        //Create Objects
        Cart TheCart = new Cart();
        Account TheAcc = new Account();
        Element Element1 = new Element();
        Element Element2 = new Element();
        Element Element3 = new Element();
        Computer Computer1 = new Computer();
        Computer Computer2 = new Computer();
        Computer Computer3 = new Computer();
        double Result = 0;

        //Give them values
        Computerdetails Computer1Details = new Computerdetails();
        Computerdetails Computer2Details = new Computerdetails();
        Computerdetails Computer3Details = new Computerdetails();

        Computer1Details.setM_Price(300);
        Computer2Details.setM_Price(200);
        Computer3Details.setM_Price(400);

        Computer1.setM_ComputerDetails(Computer1Details);
        Computer2.setM_ComputerDetails(Computer2Details);
        Computer3.setM_ComputerDetails(Computer3Details);

        Element1.setM_Computer(Computer1);
        Element2.setM_Computer(Computer2);
        Element3.setM_Computer(Computer3);

        Element1.setM_CountOfComputers(3);
        Element2.setM_CountOfComputers(2);
        Element3.setM_CountOfComputers(5);

        TheCart.getM_firstElement().setM_nextElement(Element1);
        TheCart.getM_firstElement().getM_nextElement().setM_nextElement(Element2);
        TheCart.getM_firstElement().getM_nextElement().getM_nextElement().setM_nextElement(Element3);

        //Call function
        Bill TheBill = new Bill(TheCart, TheAcc,  discount: 25,  TheTaxes: 16);
        Result = TheBill.getM_Price();

        //Compare values
        assertEquals(Result,  actual: 2871.0);


    }
```

```
@Test
void searchElement() {

//Create Objects
    Computer TheComputer = new Computer();
    Element Element1 = new Element();
    Element Element2 = new Element();
    Element Element3 = new Element();
    Element Result = new Element();
    Cart TheCart = new Cart();
    Cart TheCartFunction = new Cart();

//Give them values
    int ElementToSearch = 5;
    Element1.setM_Computer(new Computer());
    Element1.getM_Computer().setM_ItemId(4);
    Element2.setM_Computer(new Computer());
    Element2.getM_Computer().setM_ItemId(5);
    Element3.setM_Computer(new Computer());
    Element3.getM_Computer().setM_ItemId(6);

    TheCart.getM_firstElement().setM_nextElement(Element1);
    TheCart.getM_firstElement().getM_nextElement().setM_nextElement(Element2);
    TheCart.getM_firstElement().getM_nextElement().getM_nextElement().setM_nextElement(Element3);

    Result= Cartfunctions.searchElement(TheCart, ElementToSearch);

    assertEquals(Element2.getM_Computer().getM_ItemId(), Result.getM_Computer().getM_ItemId());
}
```

Was folgt?

- Anbindung an eine Datenbank
- Grafische Benutzeroberfläche
- Hinzufügen weiterer Funktionen

Lessons Learned

- Einhalten von Terminen, sowie regelmäßigen Treffen sind wichtig
- GitHub ist zwar praktisch, kann aber auch Probleme bringen
- Einheitlicher Codestyle ist wichtig
- Kleinigkeiten beanspruchen auch Zeit

Vielen Dank für eure Aufmerksamkeit!