BORD-Festival-Management-System
Java 2 - Zwischenpräsentation

Olga Klassen, Benjamin Swarovsky, Raphael Freybe, Franziska Schmidt, Daniel Depta

~~Fachhochschule Erfurt~~ → zu Hause, 15.06.2020

ALLGEMEINES

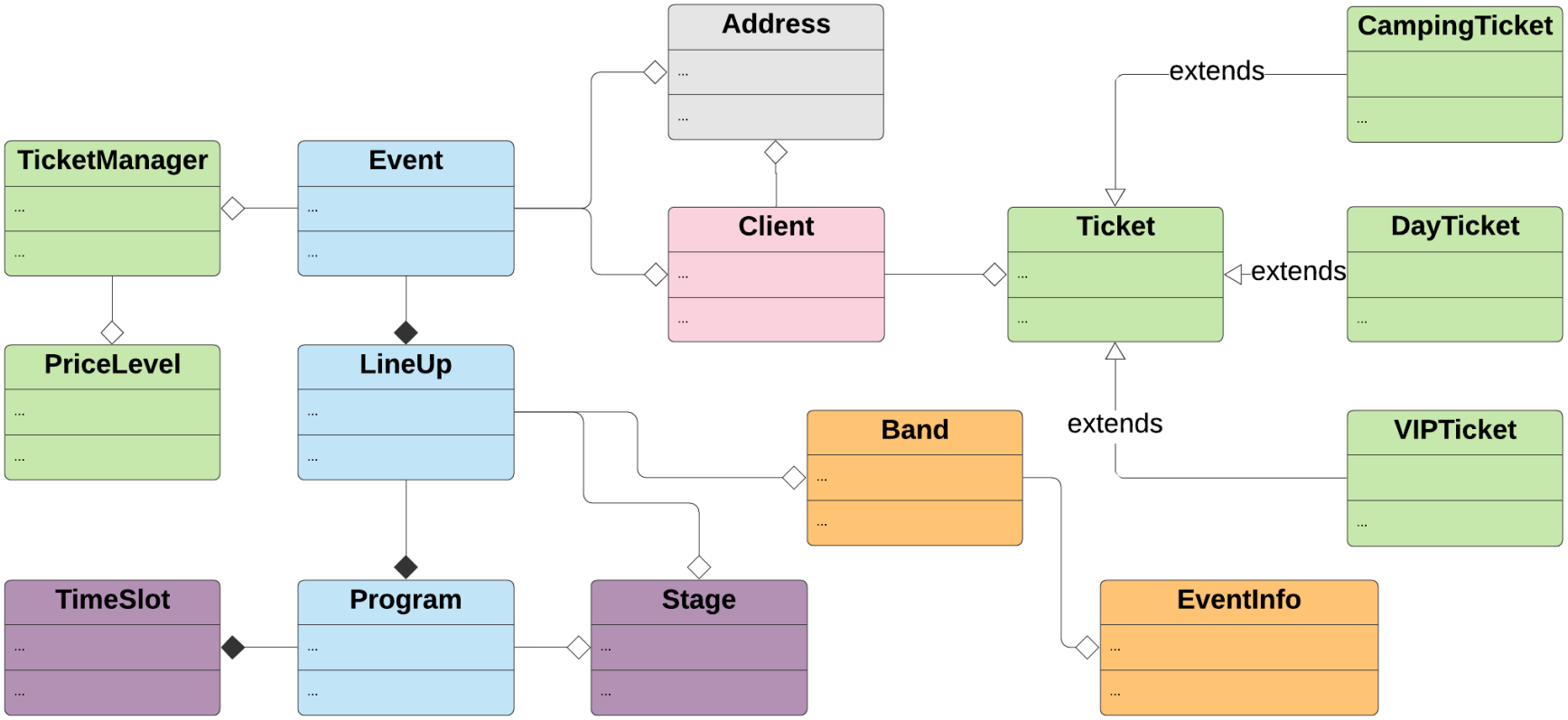
Letztes Semester:

- Festival-Management-Backend

Dieses Semester:

- Team erweitert
- Spring Boot MVC
- Java + Datenbank in der Cloud
- Dynamische Website

UML-DIAGRAMM



MIT DER EINFÜHRUNG VON JPA

```java
@Entity
public class AbstractModel {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    protected Long id;

    public Long getId() { return this.id; }

    @Version
    private Long version;

    @Temporal(TemporalType.TIMESTAMP)
    private Date createdAt;

    @Temporal(TemporalType.TIMESTAMP)
    private Date updatedAt;

    @PrePersist
    void onCreate() { this.createdAt = new Date(); }
```

```java
public abstract class AbstractRepository<T extends AbstractModel> {

    private DataSource dataSource;

    public AbstractRepository() { dataSource = DataSource.getDataSource(); }

    protected abstract void updateOperation(T model, String argument);


    public Long create(T model) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
        entityManager.persist(model);
        entityManager.flush();
        entityManager.getTransaction().commit();
        return model.getId();

    }

    public void update(T model, String argument) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
```

MIT DER EINFÜHRUNG VON JPA

```java
@Entity
public class TicketManager extends AbstractModel

    @OneToMany(
            mappedBy = "actualTicketPrices"
    )

    private List<PriceLevel> priceLevels;

 @Transient private DayTicket dayTicket;
 @Transient private CampingTicket campingTicket;
 @Transient private VIPTicket vipTicket;
```

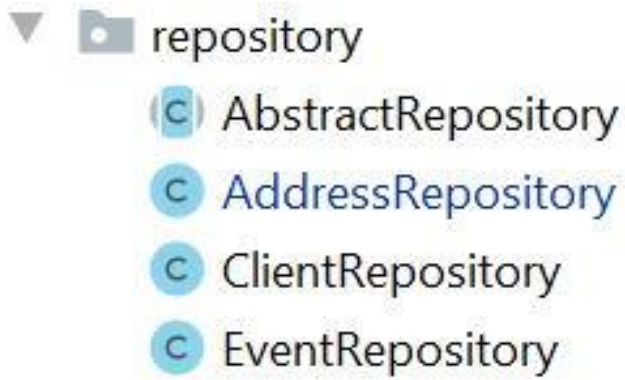
```java
@Entity
public class PriceLevel extends AbstractModel implements Comparable

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Long id;
   private double dayTicketPrice;
   private double campingTicketPrice;
   private double vipTicketPrice;

    @ManyToOne
   private TicketManager actualTicketPrices;
```

TESTEN

Main Verzeichnis



Test Verzeichnis

TESTEN

```java
public class AddressJPATest {
    AddressRepository addressRepository;
    Address address;

    @BeforeEach
    void initialize() {
        this.address = new Address( country: "Germany",  city: "Berlin",  street: "Plumb",  zip: "2211");
        this.addressRepository = new AddressRepository();
    }

    @Test
    void should_create_new_address_in_database() {
        addressRepository.create(address);
        Address databaseAddress = addressRepository.findOne(this.address);
        assertEquals( expected: "Germany", databaseAddress.getCountry());
        assertEquals( expected: "Berlin", databaseAddress.getCity());
        assertEquals( expected: "Plumb", databaseAddress.getStreet());
        assertEquals( expected: "2211", databaseAddress.getZip());

        assertEquals( expected: 1, databaseAddress.getId());

    }
```

TESTEN

```java
@Entity
public class Client extends AbstractModel implements IClient {

    @ManyToOne(cascade = CascadeType.PERSIST)
    private Address address;

    private String firstName;
```

```java
public class ClientJPATest {
    ClientRepository clientRepository;
    Client client;

   @BeforeEach
    void initialize() {
        HelpClasses helper = new HelpClasses();
         this.client = new Client( firstName: "Max",  lastName: "Mustermann",  mail: "max.muster@mann.de", helper.getAddress());
         this.clientRepository = new ClientRepository();
    }
```

TESTEN

```
@Test
void should_create_new_client_in_database() {
    clientRepository.create(client);
    Client databaseClient = clientRepository.findOne(this.client);  databaseCli
    assertEquals( expected: "Max", databaseClient.getFirstName());  databaseClient
    assertEquals( expected: "Mustermann", databaseClient.getLastName());
    assertEquals( expected: "max.muster@mann.de", databaseClient.getMail());
```

```
databaseClient = {Client@3040}
  address = {Address@3041}
    country = "Germany"
    city = "Berlin"
    street = "Nordwez 1"
    zip = "8803"
```

MIT DER VERWENDUNG VON SPRING

```java
public abstract class AbstractRepository<T extends AbstractModel> {

    private DataSource dataSource;

    public AbstractRepository() { dataSource = DataSource.getDataSource(); }

    protected abstract void updateOperation(T model, String argument);


    public Long create(T model) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
        entityManager.persist(model);
        entityManager.flush();
        entityManager.getTransaction().commit();
        return model.getId();

    }

    public void update(T model, String argument) {
        EntityManager entityManager = dataSource.getEntityManager();
        entityManager.getTransaction().begin();
```

DERZEITIGER STAND

```java
import org.springframework.data.repository.CrudRepository;

public interface PriceLevelRepository extends CrudRepository<PriceLevel, Long> {

    PriceLevel findById(long id);
}
```

```java
public static void main(String[] args) {
  SpringApplication.run(AccessingDataJpaApplication.class);
}

@Bean
public CommandLineRunner demo(CustomerRepository repository) {
  return (args) -> {
    // save a few customers
    repository.save(new Customer("Jack", "Bauer"));
    repository.save(new Customer("Chloe", "O'Brian"));
    repository.save(new Customer("Kim", "Bauer"));
    repository.save(new Customer("David", "Palmer"));
    repository.save(new Customer("Michelle", "Dessler"));
```

LESSONS LEARNED IN CORONA

- min. 1x wöchentliches Discord Teammeeting
- bei Problemen Bildteilung
- Zwischenkontakt über Messenger
- regelmäßiges pushen und pullen
- Issues für Aufgabenteilung
- einzelnes Lösen der Übungsaufgaben mit anschließender Teambesprechung
- voreilige Lösungen

Vielen Dank für Ihre Aufmerksamkeit!

https://github.com/fh-erfurt/bord-festival