

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Objektovo-orientované programovanie

**“Citizen Pulse”**

Záverečná dokumentácia

Illia Pukhalskyi

Čas cvičenia: Streda 12.00-13.50

Cvičaci: Mgr. A. Považanová

Datum:13.05.2024

**Opis: "Citizen Pulse: systém riadenia hlasovania pre účasť občanov na rozhodovaní".**

Tento projekt poskytne obyvateľom jednoduchý a pohodlný spôsob účasti na rozhodovaní o dôležitých otázkach v oblasti bývania a verejných služieb, ako aj sledovanie a analýzu výsledkov hlasovania.

**Funkcionalita systému Citizen Pulse zahŕňa:**

- 1. Zaregistrujte sa ako správca alebo člen a prihláste sa ako používateľ.**
- 2. Zobrazte si dostupné podujatia v rôznych oblastiach verejných služieb.**
- 3. Na stránke správcu:**
  - **Môžete pridávať a odstraňovať projekty, spravovať údaje a odstraňovať členov.**
  - **Zobrazenie výsledkov hlasovania s vizualizáciou údajov.**
  - **Zobrazenie a analýza výsledkov predchádzajúcich hlasovaní.**
- 4. Na strane účastníkov:**
  - **Hlasujte o preferovaných možnostiach projektov.**
  - **Poskytovať spätnú väzbu a hodnotiť priebeh projektu.**

## Objektovo orientovaný prístup:

### 1. Uplatnenie objektovo-orientovaných mechanizmov:

- Trieda Event: Využíva interakciu objektov vrátane deduplikácie, polymorfizmu a zapuzdrenia na správu udalostí a ich údajov.
- Trieda Participant : Interakcia s objektmi udalostí s uvedením príkladov agregácie a zapuzdrenia.
- Trieda Admin: Používa mechanizmus dedenia na prístup k administratívnym funkciám systému.

### 2. Použitie návrhových vzorov:

- Trieda Participant: Implementuje vzor pozorovateľa na oznamovanie zmien v udalostiach.
- Trieda VotingManager: Implementuje strategický vzor na správu rôznych hlasovacích algoritmov.

### 3. Ošetrovanie mimoriadnych stavov vlastnými výnimkami:

- Trieda InvalidPasswordException Táto trieda obsahuje vlastné výnimky, ktoré sa aktívne používajú na spracovanie chýb v systéme.

### 4. Poskytnutie grafického používateľského rozhrania:

- Trieda GUIUtils: Táto trieda predstavuje grafické používateľské rozhranie oddelené od hlavnej obchodnej logiky systému.
- Triedy Admin, Participant, User, EventHandler: Táto trieda implementuje obsluhu udalostí pre interakciu používateľa s rozhraním.

### 5. Explicitné použitie viacnítovosti:

- Trieda Participant: Používa viacvláknový mechanizmus na počítanie hlasov pri paralelnom hlasovaní.

### 6. Použitie generickosti vo vlastných triedach:

- Trieda LinkedList : Implementuje všeobecný zoznam, ktorý môže uchovávať prvky rôznych typov.

### **7. Explicitné použitie RTTI:**

- Trieda User : Používa mechanizmus RTTI na dynamickú analýzu a zmenu tried (Admin alebo Participant) za behu.

### **8. Použitie vnhnezdených tried a rozhraní:**

- Trieda Admin , Participant, EventHandler : Obsahuje príklad vnorenej triedy, ktorá sa používa na logické zoskupenie súvisiacich operácií.

### **9. Použitie lambda výrazov alebo referencií na metódy:**

- Trieda Participant: Demonštruje použitie lambda výrazov na vytvorenie anonymných funkcií v systéme.

### **10. Použitie implicitnej implementácie metód v rozhraniach:**

- Triedy Participant , Admin , Event: implementujú rozhranie pomocou implicitnej implementácie predvolených metód.

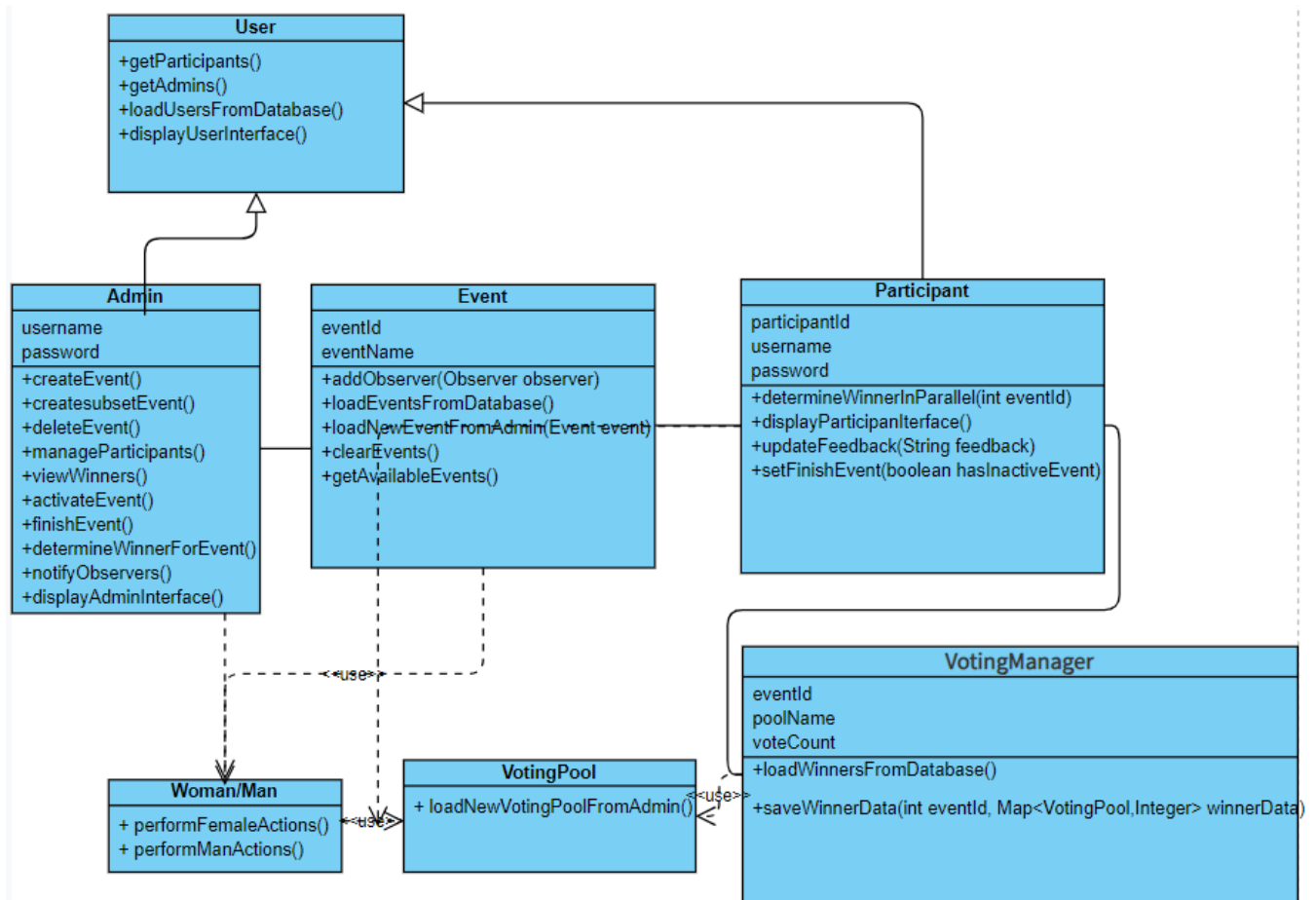
### **11. Použitie serializácie:**

- Trieda EventSerialisation: Používa mechanizmus serializácie na ukladanie a obnovovanie stavu objektov v systéme.

### **12. Použitie aspektovo-orientovaného programovania:**

- Trieda AspectJManager: Využíva aspektovo-orientované programovanie na rozdelenie cezreznej funkcionality systému.

## Štruktúra systému vo forme diagramu najdôležitejších tried s vysvetlením tried



### Admin (Administrátor):

Administrátor zohráva kľúčovú úlohu v riadení systému hlasovania. Má práva prístupu do administrátorskej konzoly, kde môže riadiť všetky aspekty systému.

Po prihlásení sa do systému pomocou metódy `login()` získa prístup k širokej škále funkcií. Metóda `createEvent()` umožňuje administrátorovi vytvárať nové udalosti, zatiaľ čo `createSubsetEvent()` dáva možnosť vytvárať podudalosti v rámci hlavnej udalosti. Po vytvorení udalosti môže administrátor riadiť účastníkov udalosti, odstraňovať udalosti pomocou metódy `deleteEvent()`, ako aj aktivovať a ukončiť udalosti pomocou metód `activateEvent()` a `finishEvent()`.

Po vykonaní všetkých potrebných operácií sa administrátor môže odhlásiť zo systému pomocou metódy `logout()`.

### Event (Udalosť):

Udalosť predstavuje hlavný prvok systému hlasovania. Obsahuje informácie o samej udalosti, ako je identifikátor a názov, ako aj zoznam pozorovateľov, ktorí sledujú jej zmeny.

Metóda `addObserver(Observer observer)` umožňuje pridať pozorovateľa pre udalosť. To umožňuje ostatným častiam systému reagovať na zmeny v udalosti. Metódy `getEventId()` a `getEventName()` vrátia identifikátor a názov udalosti. Metódy `loadEventsFromDatabase()` a `loadNewEventFromAdmin(Event event)` sa používajú na načítanie a vytváranie nových udalostí z databázy a administrátorskej konzoly.

### **Participant (Účastník):**

Účastník systému predstavuje koncového používateľa, ktorý sa zúčastňuje hlasovania. Účastník má identifikátor, užívateľské meno a heslo na vstup do systému.

Účastník sa môže zúčastniť udalosti pomocou metódy `participate(Event event)` a potom hlasovať pomocou metódy `vote(Event event)`. Metóda `determineWinnerInParallel(int eventId)` sa používa na určenie víťaza v paralelnom vlákne. Metóda `displayParticipantInterface()` poskytuje rozhranie pre interakciu s účastníkom, zatiaľ čo `setFinishEvent(boolean hasInactiveEvent)` nastavuje stav ukončenia udalosti.

### **VotingManager (Manažér hlasovania):**

Manažér hlasovania zodpovedá za riadenie hlasovania v rámci konkrétnej udalosti. Obsahuje informácie o udalosti, názve hlasovania a počte hlasov.

Metódy `loadWinnersFromDatabase()` a `saveWinnerData()` sa používajú na načítanie dát o víťazoch z databázy a ich uloženie.

### **Woman (Žena) / Man (Muž):**

Žena a muž sú podtriedy účastníka a poskytujú akcie špecifické pre pohlavie, ako sú `performFemaleActions()` a `performMaleActions()`.

**Každá z týchto tried komunikuje medzi sebou a s ostatnými triedami systému, čo zabezpečuje plynulý chod systému hlasovania.**