SPŠE JEČNÁ
IT – Programovaní a digitální technologie
Praha 2, Ječná 30

E-shop

Kohlík Richard
Informační a komunikační technologie

2025

Obsah

1 Cíl práce

Cílem tohoto projektu je vytvořit konzolovou simulaci e-shopu, která poskytuje interaktivní rozhraní jak pro běžné uživatele, tak pro administrátory. Pro uživatele je cílem umožnit procházení dostupných produktů, přidávání zboží do košíku, úpravu obsahu košíku a následné dokončení nákupu vyplněním objednávky, kde uživatel zadá údaje jako jméno, adresu a způsob platby. Po úspěšném vyřízení objednávky by uživatel měl obdržet zakoupené produkty do svého skladu. Na druhé straně, administrátoři mají k dispozici rozhraní, které jim umožňuje prohlížet produkty, spravovat stav zásob (doplňovat je) a vyřizovat uživatelské objednávky, což znamená změnu jejich stavu (konkrétně. na "odesláno" nebo "zrušeno"). Také by v programu mělo fungovat ukládání všech dat (uživatelských účtů, jejich košíků, vlastněných produktů, historie objednávek a také stav skladu), aby informace zůstaly zachovány i po restartu aplikace.

2 Popis programu

Tato aplikace představuje konzolovou simulaci e-shopu, jejíž interakce probíhá prostřednictvím textových příkazů. Program je navržen s důrazem na flexibilitu, např. dostupné funkce se mění v závislosti na aktuálně přihlášeném uživateli (běžný uživatel – administrátor).

2.1 Mechaniky programu

Program využívá návrhový vzor Command pro zpracování uživatelských vstupů, což zajišťuje přehlednou strukturu pro jednotlivé funkce. Po spuštění aplikace je uživatel nejprve vyzván k přihlášení, které určuje jeho oprávnění a sadu dostupných příkazů:

- Uživatelské rozhraní: Běžní uživatelé mají přístup k funkcím umožňujícím prohlížení produktů, přidávání zboží do košíku, odebírání položek z košíku, dokončení nákupu prostřednictvím procesu “pokladny”, zobrazení aktuálně vlastněných produktů a kontrolu historie svých objednávek.
- Administrátorské rozhraní: Administrátoři mají odlišnou sadu příkazů, které jim umožňují spravovat e-shop. Mohou doplňovat zásoby produktů, prohlížet produkty (stejně jako uživatel) a hlavně měnit stavy uživatelských objednávek, podle čeho se následně určí, jestli se objednávka doručí, nebo jestli byla zrušena.

2.2 Ukládání dat

Jedním z důležitých aspektů aplikace je schopnost ukládat data. Všechny důležité údaje, jako je obsah uživatelského košíku, seznam vlastněných produktů a historie objednávek, jsou pro každého uživatele ukládány a načteny dle toho, jaký uživatel se přihlásí (tzn. každému uživateli se ukládají data separátně a nemíchají se). Stejně tak je důležité ukládání a aktualizování celkového stavu skladu (počet produktů), což zajišťuje, že informace o dostupnosti produktů jsou vždy aktuální i po restartu aplikace (Stav skladu je pro všechny uživatelé stejný, což znamená, že když nějaký uživatel vykoupí kompletně nějaký produkt, tak jiný uživatel vidí, že není žádný na skladě). Jak bylo zmíněno předtím, program podporuje existenci více uživatelů současně a data každého uživatele jsou spravována nezávisle.

3 Systém requirements

Program byl vyvíjen v jazyce Java, konkrétně ve verzi Java SE 11. Pro správné spuštění aplikace je doporučeno používat nejnovější verzi JDK 11 nebo vyšší. Kromě samotného JDK nejsou k chodu programu potřeba žádné externí knihovny ani rámce.
Veškerá funkcionalita je řešena pomocí standardních knihoven Javy. Program je možné

spustit jak v běžném příkazovém řádku, tak v libovolném vývojovém prostředí podporujícím Javu, například IntelliJ IDEA, Eclipse či NetBeans. Důležité je také zajistit, aby prostředí mělo možnost pro zápis a čtení do složky, kde je aplikace spuštěna, neboť program ukládá uživatelská data a stav skladu do souborů (konkrétně do složky users a souboru stock.serialization). JUnit je pouze pro vývoj a testování kódu a není nutný pro spuštění aplikace.

4 Základní struktura

Architektura projektu je rozdělena do několika balíčků, což zlepšuje přehlednost tříd. Každý balíček se zaměřuje na specifickou oblast funkcionality:

- Commands: Tento balíček obsahuje všechny implementace příkazů, které může uživatel nebo administrátor zadávat v konzoli. Každý příkaz implementuje rozhraní Command, což umožňuje jednotné zpracování vstupů a dynamickou změnu dostupných příkazů na základě přihlášené role.
- Console: Zde se nachází logika pro samotnou interakci uživatele prostřednictvím konzole, včetně třídy Console, která řídí tok programu, zpracování příkazů a přepínání mezi uživatelským a administrátorským rozhraním. Nachází se zde také třída Data, která se stará o ukládání dat (uživatelských údajů a stavu skladu) pomocí Java serializace, a třída User, reprezentující uživatelský účet.
- Store: Tento balíček spojuje třídy reprezentující základní entity e-shopu. Zahrnuje třídu Product (a její podtřídy Food, Clothes, Electro pro konkrétní typy produktů), třídu Order pro správu objednávek a třídu Stock, která reprezentuje aktuální stav všech produktů na skladě.
- AbsoluteUnit: Obsahuje testovací třídy. Konkrétně se zde nachází třída Probe, která využívá framework JUnit k ověřování klíčových funkcionalit aplikace, jako je správná serializace a deserializace uživatelských dat, tok procesu od vložení do košíku po odeslání objednávky a aktualizace vlastněných položek. Testy jsou navrženy tak, aby zajistily správnost a spolehlivost nejdůležitější částí programu.

Program kombinuje dva způsoby načítání dat: počáteční stav skladu je načítán z textového souboru, zatímco uživatelská data a aktuální stav skladu jsou ukládány a načítány pomocí serializace. Tato hybridní metoda je výhodná pro snadnou inicializaci, tak i ukládání dynamických dat.

Hlavními řídícími třídami a oblastmi, kde se koncentruje složitější logika a interakce mezi všemi třídami, jsou Console a Login pro správu přihlášení a rolí, a pak zejména Checkout pro zpracování objednávky s validací uživatelských vstupů.

5 Testovací data

Pro zajištění správné funkčnosti aplikace byly implementovány jak automatizované testy, tak i doporučené manuální testovací scénáře. Cílem je odhalit potenciální chyby a ověřit, zda program reaguje očekávaným způsobem na různé uživatelské vstupy a zdali poskytuje zpětnou vazbu.

5.1 Automatizované testy (JUnit)

Pro důležité části kódu byly vytvořeny unit testy. Tyto testy se zaměřují především na ověření funkčnosti mechanismů aplikace jako:

- Serializace a deserializace uživatelských dat: Testuje se, zda se uživatelské účty, včetně jejich košíku, vlastněných produktů a historie objednávek, správně ukládají na disk a zda je možné je načíst zpět.
- Proces vytvoření objednávky: Zde se ověřuje se celý cyklus objednávky – od

přidání produktu do košíku, přes vytvoření objednávky, až po doručení produktu uživateli (změna stavu objednávky administrátorem a následné přenos zboží uživateli po následném přihlášení).

- Správa stavu skladu: Testy kontrolují, zda se množství produktů na skladě snižuje po nákupu a zvyšuje po doplnění administrátorem. Také se ověřuje funkčnost vyhledávání produktů ve skladu.

5.2 Nápady na manuální testování programu

- Přístup k příkazům: Zkontrolovat, jestli uživatelé a administrátoři mají přístup pouze k příkazům určeným pro jejich roli.
- Nákup zboží: Vyzkoušet nákup zboží, které není skladem.
- Změna stavu objednávky (administrátor): Vytvořit několik objednávek a jako administrátor je zkusit nastavit na "shipping". Následně ověřit, jestli se "shipping" objednávka objeví ve "stashi" uživatele po jeho dalším přihlášení.
- Chybné vstupy v checkoutu: Během procesu checkoutu zkusit zadat neplatné formáty pro jméno, adresu, číslo karty (např. písmena místo čísel, háčky/čárky).
- Chybné vstupy obecně: Zkusit kdekoli zadat nějaký nežádoucí vstup a ověřit, jestli program nespadne a poskytne zpětnou vazbu.

6 Uživatelská příručka

Ovládání programu probíhá prostřednictvím textové konzole, kde uživatel zadává konkrétní příkazy. Interakce je navržena tak, aby byla přehledná a poskytovala relevantní možnosti v závislosti na aktuálním stavu a roli uživatele.

6.1 Základní ovládání

Po spuštění programu se uživatel nejprve dostane k výzvě přihlášení. Zde musí zadat příkaz „login“ a následně své 8místné ID (pro administrátora je předdefinováno ID 00000000). Po úspěšném přihlášení se změní sada dostupných příkazů.

Program by měl vždy po provedení příkazu, nebo po chybě, vyhodit zpětnou vazbu. K ukončení programu slouží příkaz exit v libovolném rozhraní.

6.2 Nápověda pomocí „help“ příkazu

Klíčovou součástí uživatelské příručky je příkaz help, který poskytuje pro uživatele nápovědu. Jeho funkcionalita se mění podle toho, zda je uživatel přihlášen, a jakou má roli:

- Před přihlášením: Zobrazení instrukcí pro přihlášení a základní informace o programu.
- Pro běžného uživatele: Výpis všech dostupných příkazů pro uživatele s krátkým popisem jejich funkce.
- Pro administrátora: Také podobný výpis všech dostupných příkazů s krátkým popisem jejich funkce.

Tato struktura zajišťuje, že uživatel má vždy k dispozici informaci o tom, co může v dané fázi programu dělat.

7 Závěr

Tvoření této konzolové simulace e-shopu přinesla několik výzev. Jedním z největších problémů, který se objevil během vývoje, bylo vytvoření odlišného rozhraní a sadu dostupných příkazů pro administrátora a běžného uživatele. Původní myšlenka byla začít s univerzálním systémem, který se později rozdělí. Nakonec jsem se rozhodnul oddělit logiku a dostupnost příkazů pro každou roli, což si vyžádalo značnou úpravu struktury

kódu. Další výzvou bylo zpracování okrajových případů, jako je nedostatek financí u uživatele při platbě kartou. Zde jsem se rozhodl (po zvážení několika variant), že v případě nedostatečných prostředků se objednávka automaticky zruší, podobně jako by ji zrušil administrátor. Tím se zjednodušila logika a předešlo se zbytečným komplikacím.
Další překážkou bylo také správné navržení toku objednávky a jejího doručení uživateli. Jako řešení jsem vymyslel, že objednávka, kterou uživatel vytvoří, se nejprve nastaví na stav "čeká na potvrzení" administrátorem a teprve po jeho potvrzení (změnou statusu na "shipping") se produkty převedou do uživatelského "stashe" (Tam, kde vidí všechny vlastněné produkty) při jeho dalším přihlášení.
Celkově projekt zlepšil mé porozumění návrhovým vzorům, zejména Command patternu, který bylo ideální pro zpracování uživatelských příkazů. Rovněž práce se serializací dat pro ukládání a načítání uživatelských účtů a stavu skladu byla užitečná a přinesla cenné zkušenosti pro to, jak ukládat objekty. Celkový dojem z práce je pozitivní, protože mi dala zkušenosti a zlepšila moji schopnost programovat jako takovou.

8 Zdroje

https://www.youtube.com/watch?v=DfbFTVNfkeI