

Semestrálna práca 1

z predmetu Pokročilé objektové technológie 2016/2017

Zadanie

Vytvorte **jednoduchú textovú hru** („textovku“), ktorá bude ovládateľná nasledujúcimi textovými príkazmi:

- **go XXX** (alebo *chod XXX*) pre presun medzi miestnosťami, pričom XXX bude definovať názov miestnosti – uskutočnenie pohybu pritom závisí od aktuálneho umiestnenia hráča v hre,
- **look** (*preskumaj*) na zobrazenie predmetov v miestnosti,
- **take XXX** (*zober predmet*), **put XXX** (*vylož predmet*), **use XXX** (*použi predmet*), kde XXX bude názov predmetu (napr. počítač, mobil, adaptér, elektrická zásuvka, ...) nachádzajúceho sa v miestnosti alebo v batohu,
- ...

Program nemusí byť zložitý, jeho obsah a možnosti ponechávam na vlastnú tvorivosť, avšak mal by byť vypracovaný s dôrazom na **objektovú architektúru** (dobře navrhnuté triedy ako napr. *Miestnosť*, *Hráč*, *Predmet*, rozhrania ako *IPohybujúciObjekt*, *IPredmet* apod.) **s využitím dedičnosti a polymorfizmu**.

Povinné požiadavky:

- Vytvorte si jedno riešenie (solution) a v ňom **aspoň dva projekty** nasledujúceho typu:
 - **Class library** – knižnica DLL **bude obsahovať vlastný jednoduchý framework, pomocou ktorého bude možné vytvoriť univerzálnu textovú hru** (t. j. knižnica bude obsahovať všetky triedy a také dátové typy, ktoré budú potrebné pre vytvorenie ľubovoľnej jednoduchej textovej hry).
 - **Console application** – **implementácia textovej hry** s vymysleným dejom a cieľom, ktorý vytvoríte prostredníctvom navrhnutých typov z prvého projektu (nareferencujte si prvý projekt a použite všeobecné triedy a iné typy vytvorené v ňom; vytvorte si napr. nové triedy, ktoré budú dediť od všeobecných z knižnice alebo vytvorte nové kompozíciou tak, aby ste mali všetky objekty potrebné pre vlastnú hru).
- **Objektová architektúra** – použite **aspoň 4 triedy** a **aspoň 1 rozhranie**, ďalej **generické kolekcie** (vstavané alebo vlastné) a v triedach **vlastnosti** a **udalosti**.
- **Ukladanie stavu hry** pomocou **serializácie** a **deserializácie do XML** (stav hry sa bude môcť uložiť, aby sme sa neskôr k nemu mohli vrátiť a v začatej hre pokračovali).
- **Práca s databázou** – **ukladajte skóre hry do databázy** (s ním ukladajte tiež meno hráča, nahraté skóre, dĺžku hry, dátum a čas pripísania skóre a názov počítača, na ktorom sa hra hrala) a umožnite ho aj zobrazíť (posledných 10 najlepších hier). Databázu môžete použiť MS SQL Server (vlastnú na lokálnom počítači alebo pridelenú na databázovom cvičení na školskom serveri – prístupové údaje ku vzdialenému serveru budú poskytnuté).
- Príkazy hry, hra samotná a zdrojový kód programu môže byť napísaný buď v slovenčine, češtine alebo v angličtine.

- **Všetky triedy, dátové členy, vlastnosti a metódy** budú obsahovať **dokumentačné komentáre** (///) a obyčajné komentáre (//) všade tam, kde je to potrebné pre pochopenie kódu.
- **Vytvorte dokumentáciu**, v ktorej uvediete:
 - stručné predstavenie zvolenej hry a príbehu,
 - UML diagram tried navrhutej architektúry hry,
 - zoznam použitých technológií,
 - požiadavky na hru,
 - problémy a riešenia, s ktorými ste sa popasovali,
 - záver – zhodnotenie úspešnosti implementácie a zhrnutie možností výslednej aplikácie, možnosti pokračovania ďalšieho vývoja apod.

Dokumentácia nebude mať formálne, gramatické a štylistické chyby!!!

- **Zdrojový kód musí byť prehľadný, dobre naformátovaný a čitateľný** (používajte vhodné zvolené názvoslovie identifikátorov a **dodržiavajte rovnakú konvenciu** v celom zdrojovom kóde pre identifikátory tried, rozhraní, dátové členy, konštanty, premenné. Ďalej pozície zložených zátvoriek nech sú v rovnakom štýle .NETu alebo Javy – dobrá pomôcka je nastaviť si formátovanie v nastavení Visual Studio a konvencie v ReSharperi, ak ho používate a nevyhovujú Vám tie, ktoré používa).
- **Ošetrte chyby programu** (program nesmie spadnúť, ani vyhodiť neošetrenú výnimku).
- Používajte konštanty (napr. zavedenie *const string identifikátor* namiesto často sa opakujúcich rovnakých reťazcov, ktoré sa používajú napr. na porovnanie, čo môže spôsobovať problémy, ak sa spraví preklep alebo sa v budúcnosti zmení iba jeden z nich a zabudne sa zmeniť druhý).
- **Nepoužívajte metódy s rovnakým alebo podobným kódom – ak také máte, refaktorujte ich** (rovnaké bloky kódu nahraďte za volanie jednej spoločnej metódy apod.), **metódy by pritom mali byť jednoduché na niekoľko riadkov**. Ak používate dlhú metódu (napr. viac ako 40 riadkov alebo viac než sa zmestí na jednu obrazovku), pouvažujte nad jej rozbitím do samostatných metód.
- **Odstráňte nepoužité premenné a usingy** (pomôžte si ReSharperom alebo iným nástrojom).
- **Ak to nie je potrebné, nevytvárajte viac solutionov** (namiesto toho používajte viac projektov v jednom solutione).
- **Odobrzaný zdrojový kód bude skompilovateľný bez chýb a upozornení** (teda žiaden warning alebo error; pokiaľ máte warning z hľadiska cesty PATH, môže to byť len na Vašom PC; ďalšie možné chyby - referencovanie DLL na taký súbor, ktorý na druhom počítači neexistuje; pozor tiež aj na cesty zdrojov - obrázkov alebo iných súborov, ktoré na Vašom PC môžu byť, ale po presune zdrojových kódov na iný počítač nebude možné kvôli chýbajúcim súborom program skompilovať).
- **Odobrzané riešenie bude obsahovať iba zdrojové kódy bez priečinkov bin a obj** (tieto priečinky nie sú potrebné pre skompilovanie, preto ich pred zbalením do ZIPu odstráňte).
- Ak nejakú časť kódu skopírujete z internetu, uvediete v komentári zdroj (URL odkaz), z ktorého ste čerpali.
- **Ak bude nájdený plagiát** a v komentároch nebudú zdroje, **automaticky získavate -100 bodov (t. j. FX z predmetu)** – kto je pritom pôvodca, nebude zisťované!

Hodnotenie

Semestrálna práca je kontrolovaná v týchto krokoch:

1. Automatická kontrola – ak úloha neprejde testom skompilovania alebo antiplagiátorským softvérom, automaticky získavate 0 bodov.
2. Manuálna kontrola – prezretie zdrojového kódu, otestovanie a spísanie pripomienok.
3. Obhajoba – po odovzdaní sú niektoré semestrálne práce vybrané pre obhajobu s možnosťou ďalšej implementácie požadovaných funkcionalít na mieste obhajoby.
4. Pridelenie bodov skombinované automatickou a manuálnou kontrolou.

Príklad hry (pre inšpiráciu)

Nachádzame sa v byte. Byt má tri miestnosti – kuchyňu, spálňu a obývačku. V každej sa nachádza nejaký predmet (v prvej notebook, v druhej adaptér k notebooku, v tretej prázdny pohár a voľná elektrická zásuvka). Cieľom hry bude pripojiť notebook ku sieti, lebo sa vybila batéria a chceme na ňom rýchlo spraviť túto semestrálnu prácu. Takže treba nájsť adaptér k notebooku a jednu voľnú elektrickú zásuvku (jedna voľná sa nachádza iba v obývačke, pričom ostatné sú obsadené a nesmú sa odpojiť, lebo nastane koniec sveta). Preto sa príkazmi presunu presunieme do miestnosti, kde zoberieme notebook, v inej adaptér a v tretej pripojíme notebook s adaptérom k zásuvke. ☺ Potom sme smädní, tak musíme nájsť pohár a v kuchyni si naberať vodu. Ak si vodu načapujeme, hra končí s úspechom, lebo sme následne vodu vypili a tá bola taká osviežujúca, že sme semestrálnu prácu spravili ľavou zadnou. ;-)

Zdroje, kde môžete hľadať ďalšie inšpirácie:

- [1] <http://www.textovky.cz/>
- [2] <https://www.root.cz/clanky/historie-vyvoje-pocitacovych-her-4-cast-zlata-era-textovek/>
- [3] <https://www.root.cz/clanky/historie-vyvoje-pocitacovych-her-137-cast-slovenske-a-ceske-textovky-iv/>
- [4] http://en.wikipedia.org/wiki/List_of_text-based_computer_games