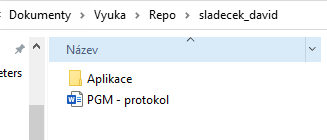
# **Zadání projektu z předmětu programování**

Úkolem bude vytvoření konzolové aplikace dle vybraného tématu. Součástí práce bude vytvoření vývojového diagramu řešící hlavní část algoritmu a vyplnění protokolu k projektu.

Konečný termín odevzdání přes aplikaci Teams je **9.1.2021**.

## **Formát odevzdání**

Odevzdávat budete archiv pojmenovaný ve formátu **prijmeni\_jmeno**. Obsahem archivu bude složka pojmenovaná opět ve formátu **prijmeni\_jmeno** a bude obsahovat složku s funkční a spustitelnou konzolovou aplikací se zdrojovými kódy a vyplněný protokol.



Obr. 1 ukázka obsahu složky

## **Požadavky na kód aplikace**

Kód aplikace bude rozumně členěn do funkcí bez zbytečného opakování kódu a okomentován (alespoň hlavní funkční části programu). Kód nehlásí chybové hlášení, je spustitelný a má určitou formu a formát. Žádný špagety kód případně rozsypané části kódu. Dbejme na čistotu kódu (lze využít automatického formátování **ctrl + k** a následně **ctrl + d**)

## **Hodnocení projektu**

Za projekt lze získat celkově **50 bodů** – 35 bodů za aplikaci, 15 bodů za protokol s vývojovým diagramem. Jakékoliv smysluplné vylepšení původního zadání případné rozšíření funkcionality je ohodnoceno až 5 bonusovými body. Princip hodnocení zůstává stejný jako u dosavadních úloh, které jste odevzdávali.

## **Známkování projektu**

Z každé části, tedy aplikace a protokolu je nutné získat minimálně 7 bodů. Projekt bude hodnocen známkou A (10).

|  |  |
| --- | --- |
| **Hodnocení (známka)** | **Minimální počet bodů** |
| 1 | 42 a více bodů |
| 2 | 29 – 41 bodů |
| 3 | 21 – 28 bodů |
| 4 | 15 – 20 bodů |
| 5 | Méně jak 15 bodů |

# **Podrobné zadání jednotlivých projektů**

Následuje podrobnější zadání jednotlivých aplikací. Při ladění předpokládáme, že uživatel nám zadává pouze správná vstupní data. Případné ošetření vstupů bude hodnoceno bonusovými body. Každá z aplikací bude mít v rozumné míře vyřešené uživatelské ovládání. Ovládání je intuitivní i bez čtení manuálu a informuje uživatele o důležitých věcech. Návrh rozhraní je součástí hodnocení. V některých případech bude třeba malý výzkum ohledně dané problematiky v případě, že jste se s daným problémem ještě nesetkali. Pokud byste měli nejasnosti případně dotazy k zadání, obraťte se na mě a zadání objasním.

1. **Řešení kvadratické rovnice v oboru přirozených čísel**

Vytvořená aplikace řeší výpočet kořenů kvadratické rovnice zadané v obecném tvaru

**Ax2 + Bx + C = 0**

Výstupem bude informace kolik kořenů má křivka zadaná kvadratickou rovnicí a jaké jsou jejich hodnoty. V případě, že zadání nebude řešitelné v oboru reálných čísel, bude uživatel o této skutečnosti informován. Pokud uživatel zadá takové vstupní hodnoty, že se nebude jednat o kvadratickou rovnici aplikace ohlásí informaci, že vstupní funkce není kvadratická.

1. **Kořeny polynomu 2. stupně metodou půlení intervalu**

Hledání kořenů polynomu druhého stupně (kvadratická rovnice) pomocí metody půlení intervalu. Vstupními hodnotami jsou koeficienty a, b, c kvadratické rovnice, interval hledání průsečíku s osou x <x,y> a přesnost s jakou budeme kořen hledat.

Princip hledání kořenu polynomu metodou půlením intervalu spočívá v rozdělení vstupního intervalu na dvě poloviny a následnému zjištění jakých hodnot nabývá funkce na okrajích intervalu a v jeho středu. Vstupní interval se dále upraví a s tím, že se hodnota jednoho z krajních bodů zamění za střed. Pro rozhodnutí, který z okrajů bude nahrazen střed využijeme faktu, že pokud se mezi body nachází průsečík, tak v jednom bodě výsledek kladný a ve druhém záporný. Takto opakovaně půlíme nově vzniklý interval až se dostaneme na dostatečně malý interval, kde se nachází průsečík.

Pokud nám při prvním počítání hodnot na obou koncích intervalu i ve středu vyjde stejné znaménko, znamená to, že se v daném intervalu nenachází průsečík a informujeme uživatele o této skutečnosti.

1. **Výpočet faktoriálu**

Aplikace, která s využitím rekurze, případně bez ní určí výsledek faktoriálu zadaného čísla. Faktoriál čísla odpovídá postupnému násobení čísla. Příklad na vstup dostaneme číslo 4, výsledek 4! (čteme 4 faktoriál) je: 4! = 4 \* 3 \* 2 \* 1 = 24. Faktoriál je definovaný pouze pro kladné celé číslo. Pokud je číslo záporné informujeme uživatele o nemožnosti vypočítat faktoriál. Stejně nelze spočítat faktoriál pro desetinné číslo. Zajímavostí faktoriálu je, že 0! = 1! = 1.

1. **Výpočet celkového odporu paralelního a sériového zapojení**

Aplikace nám umožní určit celkový odpor sériového a paralelního zapojení rezistorů. Od uživatele získáme informaci, zda budou rezistory zapojené sériově nebo paralelně a následně získáme neznámý počet odporů. Způsob načtení rezistorů je na vás. Následně po zadání všech hodnot odporů zobrazíme výsledek. Zadané odpory nesmí být záporné ani nulové.

1. **Řešení soustavy lineárních rovnic o dvou neznámých**

Aplikace nám pomůže vyřešit úlohu dvou rovnic o dvou neznámých. Na vstup získáme obě rovnice – formát vstupu nechám na vašem uvážení a následně určí výsledek. Bonusové body lze získat, pokud se výsledku budou zobrazovat jako zlomky ve tvaru 3/8 apod.

1. **Podobnost trojúhelníků**

Aplikace načte potřebné informace o dvou trojúhelnících a následně rozhodne, zda jsou trojúhelníky podobné. Pokud uživatel zadá taková data, které nám neumožní zkonstruovat trojúhelník je uživatel informován o neúspěchu případně bude vyzván k opakovanému zadání vstupních hodnot.

Na výstup zobrazujme informaci, zda jsou vstupní trojúhelníky podobné, či nikoliv.

1. **Určení průměru, mediánu, prvního a třetího kvartálu řady hodnot**

Úkolem aplikace bude určit základní statistické údaje o souboru hodnot, který získáme od uživatele. Vstupem bude řetězec obsahující všechny datové hodnoty. Jednotlivé položky budou odděleny pomocí středníku (;). Ze vstupních určíme a vypíšeme hodnoty pro medián, aritmetický průměr, první a třetí kvartál hodnot. Další statistické hodnoty souboru budou ohodnoceny bonusovými body navíc.

1. **Převod mezi číselnými soustavami (binární, decimální, hexadecimální)**

Aplikace, která nám umožní převod mezi číselnými soustavami. Nejprve si zvolíme typ převodníku   
10->16, 2->16 apod a následně od uživatele získáme číslo, které budeme chtít převést. Před samotným převedením do požadované soustavy zkontrolujeme, zda je zadané číslo opravdu zapsané v požadované soustavě. V případě, že není bude uživatel požádán o nové zadání. V opačném případě zobrazíme číslo v požadované soustavě. Bonusové body lze získat za přidání převodu do osmičkové a dalších číselných soustav.

1. **Převod mezi obloukovou a stupňovou mírou úhlů**

Pro práci s úhly používáme buď stupně případně radiány. Vytvořená aplikace nám umožní převod z radiánů na stupně. Stupně budou zapisovány s přesností na sekundy jak na vstupu, tak na výstupu. Např.: 12° 48‘ 12‘‘. Od uživatele si vyžádáme informaci, z jakých jednotek převádíme a následně vstupní hodnotu převedeme a vypíšeme na výstup.

1. **Průměrná rychlost vozu, který přejíždí symetrický kopec**

Vozidlo přejíždí dva symetrické kopce (stejná délka pro stoupání i klesání). Úkolem aplikace bude určit průměrnou rychlost vozidla, které tyto kopce přejíždí. Od uživatele si vyžádáme průměrné rychlosti stoupání a klesání na jednotlivých kopcích. Rychlost nesmí být záporná. V momentě, kdy máme načtené rychlosti určíme průměrnou rychlost pohybu a zobrazíme ji uživateli.

1. **Coulombův zákon**

Coulombův zákon řeší přitažlivost a odpudivost dvou těles. Od uživatele si vyžádáme potřebné údaje o obou tělesech nutné pro výpočet a následně zobrazíme sílu jakou na sebe tělesa působí a zde se přitahují případně odpuzují. Konstanty použité ve vzorcích budou skutečně konstanty, které uložíme do speciální proměnné. Bonusové body lze získat, pokud proměnnou definujeme jako konstantu.

1. **Určení století na základě roku**

Aplikace, která na základě zadaného data, datum může být i před naším letopočtem, určí, jaké století odpovídá danému datu. Uživatel zadá na vstup celé datum např.: 12. 8. 860 př. n. l. a výstupem bude století daného datumu, v tomto případě 9. století před naším letopočtem.

1. **Převody datových jednotek v počítači (k, M, G, T bit/byte)**

Aplikace umožňující převody jednotek používaných v oblasti informačních technologií. Uživatel zadává na vstup datovou velikost a jednotku požadovanou na výstupu. Velikosti nesmí být záporná. Např.: 18kiB kb aplikace nám z tohoto vstupu převede hodnotu 18 kilobytů na kilobity. Bonus lze získat za vypsání vstupní hodnoty ve všech ostatních jednotkách mimo zadanou vstupní hodnotu.

1. **Trojčlenka pro přímou a nepřímou úměru**

Aplikace, která nám umožní vyřešit úlohu s procenty za použití přímé nebo nepřímé úměry a trojčlenky. Pomocí této aplikace lze určit libovolnou z chybějících hodnot, kterou potřebujeme k výpočtu trojčlenky – vždy je jedna z hodnot neznámá. Pokud by během výpočtu mělo dojít k dělení nulou, bude uživatel informován o chybě. Vstupní hodnoty budou pouze kladná čísla.

1. **Kalkulačka úhlů**

Kalkulačka, která nám umožňuje vykonávat operace sčítání, odčítání, násobení a dělení úhlů. Pokud chceme použít sčítání nebo odčítání úhlů, na vstup od uživatele získáme dva úhly ve stupních (12° 3‘), pokud využíváme násobení případně dělení na vstup získáme úhel a celé číslo, které použijeme ve výpočtu. Pokud uživatel zadá špatný formát úhlu bude o tomto faktu informován.

1. **Kontrola správného počtu závorek v matematickém příkladě**

Při řešení matematického kalkulu musíme dodržovat správný počet závorek a jejich umístění, abychom mohli při výpočtu správně určit pořadí jednotlivých operací. Úkolem aplikace bude načíst matematický zápis příkladu, který bude kontrolovat, zda jsou správně zapsané závorky ve výrazu. Například (12 +3) \* (1 / 5). Naopak výraz (12 \* (() 4 + 5)) označí jako špatně zapsaný.

1. **Počet výskytů jednotlivých znaků abecedy ve větě**

Zadáním aplikace, je z načteného řetězce spočítat počet výskytů znaků abecedy ve vstupní větě. Například z řetězce „aab cdc“ získáme informaci a - 2, b – 1, c -2, d - 1. Formát výpisu nechám na vašem uvážení. Bonusové body získáte za seřazení jednotlivých znaků od nejčetnějšího znaku s nejnižším výskytem. Do výpisu nezahrnujeme mezery, čárky apod.

1. **Výpočet obsahu trojúhelníku bez znalosti výšky**

Úkolem aplikace bude určení obsahu libovolného trojúhelníku bez znalosti některé z výšek trojúhelníku. Před samotným výpočtem obsahu provedeme kontrolu, zda ze zadaných hodnot lze sestrojit trojúhelník. V případě, že nikoliv informujeme uživatele a vyzveme jej k novému zadání hodnot.

1. **Zjištění, zda je zadané číslo prvočíslem**

Aplikace, která nám umožní rozhodnout, zda číslo, které jsme zadali, je prvočíslem. Existuje několik možných způsobů, jak toto rozhodnout. Bonusové body lze získat za optimalizaci výpočtu. Vstupní hodnota nesmí být záporná.

1. **Určení barvy světla na základě vstupní vlnové délky a její barevné zobrazení**

Barva světla lze zjistit na základě vlnové délky. Vstupní hodnotou je tedy vlnová délka od uživatele. Pokud je hodnota, kterou zadáme neplatná nebo mimo rozsah, vyžádáme si opakování vstupu. Podle zadané vlnové délky vypíšeme, o jakou barvu se jedná a tuto barvu zobrazíme. Například změnou barvu textu případně pozadí konzole. Bonus lze získat, pokud výslednou hodnotu zobrazíte i v hexadecimálním zápisu.

1. **Síla potřebná ke zvednutí tělesa o určité hmotnosti na kladkostroji složeného z jedné pevné a tří volných kladek**

Představme si situaci, kdy máme kladkostroj, složený ze tří volných a jedné pevné kladky. Úkolem bude určit výslednou sílu, která je nutná pro zdvihnutí tělesa o určité hmotnosti. Vstupem získáme kladnou hodnotu hmotnosti tělesa a na výstup zobrazíme výslednou sílu v Newtonech, kterou potřebujeme ke zdvihnutí. Bonusové body lze získat za možnost variabilního počtu volných a pevných kladek kladkostroje.

1. **Ohmův zákon**

Aplikace, která nám umožní určení neznáme hodnoty napětí, proudu nebo odporu ze vztahu platného pro Ohmův zákon. Od uživatele zjistíme, kterou hodnotu chce vyjádřit a následně načteme veličiny nutné k výpočtu. Veškeré vstupní hodnoty musí být kladné (nutné ověřit). V momentě, kdy uživatel zadá neplatný vstup je informován a opakovaně požádán o zadání platné hodnoty. Jakmile máme potřebné hodnoty zobrazíme výsledek požadované chybějící veličiny.

1. **Platnost 1. a 2. Kirchoffova zákonu**

Aplikace, která nám kontroluje, zda zadaný elektrický obvod splňuje náležitosti vyjádřené pomocí Kirchofových zákonů. V první řadě od uživatele získáme informaci, kterou z uvedených pouček chceme kontrolovat. Na základě volby pak načteme požadované hodnoty. V případě, že obvod splňuje náležitosti zákonu informujeme uživatele o dodržení zákonu v opačném případě rovněž informujeme o stavu obvodu.  
1. Kirchofův zákon – vstupní hodnoty napětí do příslušného uzlu uvádíme bez zneménka, hodnoty napětí vycházející z uzlu označujeme záporným znaménkem.

2. Kirchofův zákon – velikosti napětí zdrojů uvádíme bez znaménka, napětí na spotřebičích se záporným znaménkem  
Bonus lze získat, za zobrazení přebytečného/chybějícího napětí pro obvod nebo uzel.

1. **Zjištění, zda je trojúhelník pravoúhlý**

Aplikace rozhoduje, zda je zadaný trojúhelník pravoúhlý. Před samotnou kontrolu pravoúhlosti se nejprve musíme přesvědčit o tom, že trojúhelník lze ze vstupů sestrojit. Vstupy nesmí být záporné. V případě zadání záporných nebo jinak neplatných hodnot informujeme uživatele o chybě v chodu programu.

1. **Výpočet potencionální sázkové výhry**

Aplikace, která nám umožňuje výpočet potencionální výhry v sázkové kanceláři na dostizích. Uživatel nám zadá kladné sázkové kurzy (používáme evropského značení) a vsazenou částku. Aplikace nám vrátí potencionální výhru, kterou může sázkař vyhrát. Bonusové body lze získat pro vytvoření pevné tabulky kurzů (cca 10 záznamů) ve formátu Jméno koně – kurz na výhru. Následně uživatel nebude zadávat kurzy, ale jména koní a částku, kterou chce vsadit. Při zadání vstupů, které nepotřebujeme neplatné hodnoty kurzů nebo při bonusu neplatná jména koní je uživatel informován.

1. **Šťastných 10**

Náročnější verze Štastných 10. Aplikace nám umožňuje vsadit čísla ve známé hře šťastných 10. Pokud uživatel zadá čísla, která se v sázce nemohou vyskytovat je o tom informován. Před tipováním čísel si vygenerujeme náhodný výherní los o deseti číslech a jedním dodatkovým číslem. Následně si vyžádáme los uživatele. Následně podle počtu uhodnutých čísel uživateli zobrazíme kolikátou výhru získal. Zobrazujeme výhru pouze pokud hráč tipl 7 a více čísel. Pokud uživatel uhádl alespoň dodatkové číslo zobrazíme mu informaci o drobné výhře.

1. **Převod parametricky vyjádřené přímky na obecnou rovnici**

Aplikace umožňuje převod rovnice uvedené v parametrickém tvaru (formát zadání kontrolujeme) do tvaru obecné rovnice přímky Ax + B = 0. V parametrickém tvaru rovnice budeme vyžadovat, aby byl parametr zadán jako písmeno t. Jakékoliv jiné písmeno nebude akceptováno a uživatel bude informován o této chybě. V případě, že přímka zadaná parametricky je rovnoběžkou s osou x nebo y informujeme o tom uživatele. Bonus lze získat za zobrazení směrnice přímky, směrového a normálového vektoru, průsečíku s osami a dalšími hodnotami, které ze zadání rovnice lze určit.

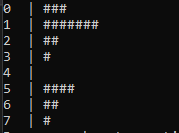
1. **Zjištění, zda je zadaný řetězec palindrom**

Palindromem nazýváme takový text, který je stejný ať jej čteme zleva doprava i obráceně. Příkladem palindromu je KobylaMaMalyBok. Aplikace rozhodne, zda zadaný řetězec je palindromem. Zadaný text nerozlišuje, zda jsou v textu zadaná malá či velká písmena případně jejich kombinace. Bonusové body lze získat, pokud dokážeme odhalit palindrom i v textu s mezerami např.: Jelenovi pivo nelej.

1. **Zjištění, zda jsou dva řetězce anagramy**

Anagramy jsou taková slova, která mají stejná písmena, ale mají jiné pořadí těchto písmen. Například z angličtiny ARMY a MARY. Tato dvě slova jsou anagramy. Aplikace nám rozhodne, zda jsou zadaná slova anagramy. Pokud nám uživatel zadá dvě stejná slova, vypíšeme tuto informaci na výstup. Aplikace nerozlišuje, zda se v textu nachází velká či malá písmena. Army a Mary budou tedy rovněž považovány za anagramy. Bonusové body lze získat, pokud anagramy odhalíme ve víceslovném textu.  
Např.: Tom Marvolo Riddle a I Am Lord Voldemort jsou anagramy.

1. **Grafické zobrazení histogramu obrazu**

Aplikace nám umožňuje zobrazení histogramu šedo tónového obrazu. Na vstupu získáme 20 hodnot reprezentující barevnou intenzitu pixelů obrazu v rozsahu 0 - 7. Histogram reprezentuje počet výskytů jednotlivých intenzit obrazu. V průběhu načítání hodnot kontrolujeme, zda načtené hodnoty jsou z požadovaného rozsahu. V momentě načtení zobrazíme histogram jako graf sloupců. Délka sloupečku bude odpovídat počtu výskytů pro konkrétní intenzitu. Jednotlivé sloupce zobrazíme v pořadí, které odpovídá intenzitě. První zobrazíme sloupec s počtem výskytů 0 a poslední sloupec s počtem výskytů 7. Pro zakreslování sloupečků využijeme symbolu **#**. Příkladem vykreslení histogramu je následující obrázek:  
 Bonusové body lze získat zobrazení histogramu ve vertikální podobě – sloupečky rostoucí od spodu nahoru.

1. **Vlastní téma**

Návrh vlastního tématu pouze se souhlasem vyučujícího a po předešlé konzultaci. Náročností by mělo téma odpovídat zhruba tématům zde uvedených.