

Domácí úloha 06

Termín odevzdání:	01.05.2023 11:59:59	11 d 15:22:23.414
Pozdní odevzdání s penalizací:	21.05.2023 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)	
Hodnocení:	5.5000 / 5.0000 (započítávat se budou všechna dílčí hodnocení)	

Vybraná / zadaná úloha: Dialogové okno I

Odevzdaná řešení: 13 / 20 + 20

Hodnocení: 3.3000 / 3.0000

Úkolem je navrhnout a implementovat sadu tříd, které budou simulovat prvky v uživatelském rozhraní. Konkrétně jde o okno (CWindow), tlačítko (CButton), textový vstup (CInput), textový popis (CL...

[Detaily](#)

Vybraná / zadaná úloha: Dialogové okno II

Odevzdaná řešení: 8 / 20 + 20

Hodnocení: 2.2000 / 2.0000

Tato úloha je rozšířením jednodušší varianty. Navíc předpokládáme zavedení prvku panel (CPanel), který umožní seskupovat ovládací prvky do skupin a s těmito skupinami pracovat. V základní variant...

[Detaily](#)

Dialogové okno I

Termín odevzdání: 01.05.2023 11:59:59

Zbývá 11 dní, 15h, 22m a 5s

Pozdní odevzdání s penalizací: 21.05.2023 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)

Hodnocení: 3.3000

Max. hodnocení: 3.0000 (bez bonusů)

Odevzdaná řešení: 13 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)

Nápovědy: 0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je navrhnout a implementovat sadu tříd, které budou simulovat prvky v uživatelském rozhraní. Konkrétně jde o okno (`CWindow`), tlačítko (`CButton`), textový vstup (`CInput`), textový popisek (`CLabel`) a combobox (`CComboBox`).

Úkol je zaměřen na návrh tříd, kde bude využito dědičnosti, polymorfismu a abstraktních metod. Pokud jsou tyto OOP prostředky použity racionálně, není implementace příliš dlouhá. Naopak, pokud provedete návrh špatně, bude se Vám kód opakovat a implementační soubor bude velký.

Navržené třídy budou GUI pouze simulovat. Nebudou nic vykreslovat, budou pouze vytvářet modely uživatelských prvků. Třídy tedy půjde použít k sestavení modelu uživatelských ovládacích prvků v okně, půjde manipulovat s jejich stavem a půjde zobrazit jejich stav ve výstupním proudu (viz níže). Jednotlivá okna a jednotlivé uživatelské prvky půjde identifikovat pomocí jejich číselného `id`, pokud toto `id` je unikátní, pak pomocí tohoto `id` lze prvek i vyhledat. Pokud `id` unikátní není, bude ovládací prvek fungovat, ale nepůjde jej podle tohoto ID vyhledat.

Okno bude reagovat na změnu umístění/velikosti. Poloha a velikost jednotlivých ovládacích prvků je zadána relativně vzhledem k šířce/výšce nadřazeného okna (velikost/poloha je tedy desetinné číslo v rozsahu 0 až 1). Při změně velikosti okna se přepočtou pozice a velikosti jednotlivých ovládacích prvků. Při přepočtu se vynásobí odpovídající šířka (výška) desetinným číslem polohy/velikosti a pro zobrazení se použije takto vzniklé desetinné číslo.

Rozhraní tříd pro reprezentaci ovládacích prvků je velmi podobné. Konstruktor typicky má 2 nebo 3 parametry. Prvním parametrem bývá celočíselné `id` přiřazené ovládacímu prvku. Následuje informace o umístění (`CRect` se 4 složkami typu desetinné číslo, které udávají relativní pozici `x`, `y` a relativní velikosti `w` a `h`). Posledním parametrem bývá řetězec, který určuje zobrazovaný obsah. Dále, všechny třídy implementující ovládací prvky musí být schopné se zobrazit ve výstupním proudu operátorem `<<` a musí být schopné se správně kopírovat (hluboké kopie). Testovací prostředí většinou požaduje pouze tuto základní funkcionalitu (tj. konstruktor, kopírování a operátor pro výstup). Vaše implementace ale zřejmě bude požadovat rozhraní další, které si do tříd budete muset doplnit.

`CButton`

Reprezentuje tlačítko. Testovací prostředí požaduje pouze konstruktor a operátor pro výstup.

`CInput`

Reprezentuje textový vstup. Posledním parametrem konstruktoru je řetězec, který je ve vstupním boxu zobrazován. Testovací prostředí požaduje konstruktor, operátor pro výstup a další dvě metody:

- `setValue (x)`, která nastaví obsah na řetězec `x`,
- `getValue ()`, která vrátí aktuálně zobrazovaný řetězec.

CLabel

Reprezentuje statický text. Posledním parametrem konstruktoru je zobrazovaný řetězec. Testovací prostředí požaduje konstruktory a operátory pro výstup.

CComboBox

Reprezentuje combobox. Testovací prostředí požaduje konstruktory, operátory pro výstup a další 3 metody:

- `add (x)`, která přidá zadaný řetězec `x` do seznamu nabídek,
- `setSelected (x)`, která nastaví nabídku s indexem `x` jako vybranou (výchozí je vybraná nabídka s indexem 0),
- `getSelected ()`, která zjistí index aktuálně vybrané nabídky.

CWindow

Reprezentuje okno. Třída má následující rozhraní:

- konstruktory `CWindow(id, title, absPos)`, který vytvoří okno s názvem `title`. Okno bude mít nastavenou pozici `x,y` a velikost `w x h` podle zadaného parametru `absPos`. Pozor, zde jsou rozměry absolutní.
- Metoda `add(x)`, která přidá ovládací prvek `x`.
- Metoda `setPosition (absPos)`, která změní pozici/velikost okna. Zadané hodnoty jsou absolutní souřadnice a velikosti okna po změně. Metoda zároveň zajistí přepočty pozice a velikosti všech ovládacích prvků obsažených v okně.
- Metoda `search (id)` vyhledá ovládací prvek podle zadaného `id`. Pokud prvek se zadaným `id` v okně neexistuje, metoda vrátí `nullptr`. Pokud v okně existuje více ovládacích prvků se stejným `id`, metoda vrátí odkaz na některý z nich (např. na první).
- Operátor `<<` zobrazí obsah okna do výstupního proudu, formát je zřejmý z příloženého zdrojového kódu.

CRect

Pomocná třída reprezentující obdélník. Tato třída je implementovaná v testovacím prostředí a je zahrnutá i do příloženého zdrojového souboru. Při odevzdávání ji ponechte v bloku podmíněného překladu.

Odevzdávejte zdrojový kód s implementací tříd `CWindow`, `CButton`, `CInput`, `CLabel` a `CComboBox`. Do odevzdávaného souboru zahrňte všechny potřebné podpůrné deklarace. Části vkládání hlaviček a Vaše testy ponechte v bloku podmíněného překladu, jak je naznačeno v příložené ukázce.

Poznámky

- Používejte operátory pro přetypování (`dynamic_cast`) s rozmyslem. V ukázce byl použit, zde mu zcela zabránit nelze. Ale vlastní referenční implementace v sobě nemá žádné přetypování ani žádné použití RTTI. Obecně, RTTI, `dynamic_cast` a `typeid` vedou k více větvenému kódu, který je hůře čitelný a hůře rozšiřitelný. Navrhněte třídy tak, abyste si vystačili s polymorfismem.
- Vaše řešení musí používat třídy, třídy musí tvořit hierarchii, dědičnost a polymorfismus musí být použité. V této úloze je použití dědičky a polymorfismu vhodné, navíc, testovací prostředí odmítne řešení, které by dědičky, polymorfismus a dynamicky vázané metody nevyužívalo (takové řešení bude odmítnuto na chybě při kompilaci).
- Výstup je ve formě "stromu", byť v této úloze je strom nejvýše dvojúrovňový. Číslo v hranatých závorkách udává `id` okna/ovládacího prvku. V kulatých závorkách jsou pak absolutní souřadnice (`x,y,w,h`).
- Výpis obsahuje přepočtené absolutní souřadnice ovládacích prvků. Absolutní souřadnice jsou známy až po vložení ovládacího prvku do nějakého okna. Testovací prostředí proto nezkouší vypisovat vytvořené objekty ovládacích prvků před jejich vložení do nějakého okna.
- Řešení této úlohy nelze použít pro code review. Lze použít až správné řešení rozšířené verze (Dialogové okno II).


Odevzdat:

 No file chosen

Referenční řešení

Hodnotitel: automat

Program zkompilován

✓	100/100	1.00	 Závažný test: Zakladni test podle ukazky	 0.001s / 2.000s
✓	100/50	1.00	 Závažný test: Test navrhů tríd	 0.000s / 1.999s
✓	100/50	1.00	 Závažný test: Test nahodnymi hodnotami	 0.028s / 1.999s
✓	100/50	1.00	 Závažný test: Test kopirujiciho konstruktora	 0.001s / 1.971s
✓	100/50	1.00	 Závažný test: Test operatoru=	 0.002s / 1.970s
✓	100/50	1.00	 Závažný test: Test nahodnymi daty + mem debugger	 0.010s / 2.000s

Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)

Celkové procentní hodnocení: 100.00 %

Bonus za včasné odevzdání: 0.30

Celkem bodů: 1.00 * (3.00 + 0.30) = 3.30

SW metriký:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	40	--	--	--
Řádek kódu:	220	5.50 ± 1.77	12	CPanel::search
Cyklomatická složitost:	51	1.28 ± 0.74	4	CPanel::search

Dialogové okno II

Termín odevzdání:	01.05.2023 11:59:59	Zbývá 11 dní, 15h, 20m a 51s
Pozdní odevzdání s penalizací:	21.05.2023 23:59:59 (Penále za pozdní odevzdání: 100.0000 %)	
Hodnocení:	2.2000	
Max. hodnocení:	2.0000 (bez bonusů)	
Odevzdaná řešení:	8 / 20 Volné pokusy + 20 Penalizované pokusy (-2 % penalizace za každé odevzdání)	
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)	

Tato úloha je rozšířením jednodušší varianty. Navíc předpokládáme zavedení prvku panel (`CPanel`), který umožní seskupovat ovládací prvky do skupin a s těmito skupinami pracovat.

V základní variantě úlohy existovalo okno, do kterého se umísťovaly ovládací prvky. Poloha ovládacího prvku tedy byla daná relativně k oknu. V praxi takový jednoduchý postup nestačí (např. pro okna, která mají posuvné příčky (splitter), kterými si uživatel může nastavit poměr plochy okna přidělený jednotlivým ovládacím prvkům). Chceme rozšířit naši implementaci, aby zvládala podobnou funkci. Tedy chceme vytvořit třídu `CPanel`, která bude schopná pojmout ovládací prvky a kterou půjde umístit do nadřazeného okna (nebo rekurentně do nadřazeného panelu). Panel se bude chovat jako speciální ovládací prvek. Jeho pozice/velikost bude definována relativně k nadřazenému oknu/panelu a velikost/poloha v něm obsažených prvků se bude přepočítávat relativně k velikosti panelu.

Testovací prostředí pro panel požaduje konstruktor (2 parametry podle ukázky), metodu `add` a operátor pro výstup. Rozhraní zbývajících tříd se nemění.

Správné řešení této úlohy, které projde povinnými testy na 100%, může být použito pro code review.

Vzorová data:

Download

Odevzdat:

Choose File No file chosen

Odevzdat

Referenční řešení

Hodnotitel: automat

Program zkompilován

✓	100/100	1.00	Závazný test: Zakladni test podle ukazky	0.001s / 10.000s
✓	100/50	1.00	Závazný test: Test navrhu trid	0.000s / 9.999s
✓	100/50	1.00	Závazný test: Test nahodnymi hodnotami (bez CPanel)	0.012s / 9.999s
✓	100/50	1.00	Závazný test: Test nahodnymi hodnotami (s CPanel)	0.311s / 9.987s
✓	100/50	1.00	Závazný test: Test kopirujiciho konstrukturu	0.011s / 9.676s
✓	100/50	1.00	Závazný test: Test operatoru=	0.011s / 9.665s
✓	100/50	1.00	Závazný test: Test nahodnymi daty + mem debugger	0.125s / 2.000s

Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)

Celkové procentní hodnocení: 100.00 %

Bonus za včasné odevzdání: 0.20

Celkem bodů: 1.00 * (2.00 + 0.20) = 2.20

SW metriky:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	40	--	--	--
Řádek kódu:	220	5.50 ± 1.77	12	CPanel::search
Cyklomatická složitost:	51	1.28 ± 0.74	4	CPanel::search