učebnice
VBalt 5.0
Visual Baltazar 5.0

Obsah

1. Úvod
1.1. Příklad jednoduchosti
1.2. Výhody visuálního programování
2. Váš první program ve VBaltu 5.0
2.1. Přidáváme tlačítko...
3. Objekty
3.1. Typy objektů
3.2. Vlastnosti objektů
3.3. Události objektů
3.4. Implicitní nastavení vlastností a událostí objektů
3.5. Přidávání nových objektů
3.5.1. Objekt NIC
3.5.2. Objekt OBRAZOVKA
3.5.3. Objekt FORM
3.5.4. Objekt TLAČÍTKO
3.5.5. Objekt TEXT
3.5.6. Objekt POSUVNÍK
3.5.7. Objekt VÝBĚR
3.5.8. Objekt PŘEPÍNAČ
3.5.9. Objekt ČASOVAČ
3.6. Přehled objektů
4. Pár rad na závěr...

A. Příloha: používání nástroje vbBMPfik
B. Příloha: ASCII tabulka ;-)

Pozn. autora: Tato učebnice není zase tak velká, aby se nedala vytisknout. Vytiskněte si ji! Zjednoduší vám to velice život.

1. Úvod

No, tak moc originální ten můj program není, no. Ale konec konců pan Bill Gates to taky nevymyslel a teď za to má miliardy, tak co?

Visual Baltazar 5.0 slouží k snadnému vytváření grafického uživatelsky příjemného prostředí pro programy pod SGP Baltazarem 5.0, tedy pod operačním systémem MS-DOS. Jistě si vzpomenete, kolik práce vám dalo, pokud jste se pokoušeli takové prostředí naprogramovat. Já jsem se tedy rozhodl udělat nástroj, který vám ušetří spousty hodin práce a přemýšlení. Bez jakékoliv námahy si za chvíli vytvoříte pěkné prostředí, které funguje, a pak se již můžete plně věnovat vlastnímu programu. Nemusíte se starat o překreslování tlačítek, jak to zařídit, aby šlo přesunovat myší okno, jak to udělat, když na některé okno kliknu, aby se přeplo nejvýš. A jak ošetřit, na které tlačítko vlastně ťukám, když je z části zakryto jiným. O toto se nemusíte starat, to jsem již vyrobil já. Vám stačí pouze naprogramovat, co se má stát, když se to tlačítko stiskne...

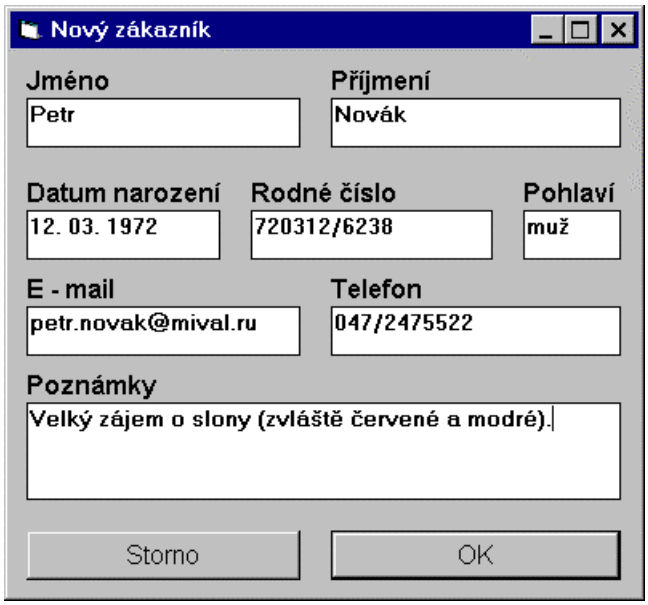
A to je právě princip visuálního programování. Nakreslíte si prostředí, nastavíte vlastnosti objektů (barvičky, velikosti, popisky a pod.) a pak již pouze ošetřujete tzv. události, které nad těmito objekty vzniknou (např. již zmiňované stisknutí tlačítka). Tedy, zatím to řeknu ještě obecně, jak se to dělá konkrétně, to vám prozradím až v dalších kapitolách.

1. 1. Příklad jednoduchosti

Dejme tomu, že si chci udělat textový editor. No, tak tam dám objekt Text. Přesně si ho umístím na obrazovku, jak chci. A už můžu editovat, jako v poznámkovém bloku... A když budu chtít ukládat, tak si tam dám ještě tlačítko, v SGP Paintu si nakreslím nějaký obrázek na něj (třeba disketu, jako ve Wordu) a pomocí nástroje vbBMPfik (součást VBaltu) si ho oříznu. No, a teď jenom napíšu prográmek, který po stisknutí tohoto tlačítka uloží text objektu Text do nějakého souboru (tento kód najdete v SGP Helperu). A je to. A jestli chci mít možnost zadávat jméno souboru, dám tam ještě jeden objekt Text a už to můžu zadávat. Ještě bych tam chtěl posuvníky. No tak je tam dám, naprogramuju, že když se pohnou, tak se pohne i pozice Textu. A je to. Mám naprogramovaný poznámkový blok, bez přemýšlení, asi tak za půl hodiny. Nebo ješte rychleji, když už to budu mít v krvi. :-)

1. 2. Výhody visuálního programování

Je to jasné, podívejte se na následující dva obrázky a hned vás to napadne, v čem je to lepší. A jestli už jste nedočkaví, jak teda můžete ve VBaltu programovat, v další kapitole vás to naučím. Uvidíte, že to je hračka. (i když teda když jsem VBalt programoval, tak to taková hračka nebyla :-)



Visuální způsob
(jeden formulář, vyplní se najednou)

```
Počítač: Nový zákazník
Počítač: =============
Počítač: zadejte jméno:
Člověk:  Petr
Počítač: zadejte příjmení:
Člověk:  Novák
Počítač: Zadejte datum narození ve
         formátu dd.mm.yyyy:
Člověk:  12.03.1972
Počítač: Zadejte rodné číslo:
Člověk:  730312/6238
Počítač: Zadejte telefon:
Člověk:  0472475522
Počítač: Napište poznámky (konec
         klávesou Esc)
Člověk:  Velký zájem o slony (zvláště
červené a modré).
Počítač: Jsou zadaná data v pořádku?
         (Ano/Ne)
Člověk:  zvoral jsem rodné číslo, takže
         musím zadat vše znova...
```

Standartní zastaralý způsob
(forma otázek a odpovědí)

2. Váš první program ve VBaltu 5.0

Aby byl tento návod přehledný, stručný a jasný, udělám to ve formě kuchařky.

1. Z adresáře VB_VZOR si zkopírujte do svého vlastního adresáře všechny 3 soubory a přejmenujte je na nový název (například POKUS).
2. V Baltazarovi otevřete soubor např. POKUS.SGP, v něm budete normálně programovat. Pokud byste ale tento program spustili, nemá zatím nikde naprogramováno, aby se ukončil. Jediné řešení je tedy ukončit pomocí Ctrl + Break.

Přidávání nových objektů:

Přepněte se na funkci __Jádro (všimněte si, že všechny vnitřní funkce VBaltu začínají podtržítkem). Přepněte se na operaci **ů** (hned na začátku skoro úplně vlevo) a editujte ji. Každý program musí obsahovat minimálně jeden objekt, a sice Obrazovka (objekt[0]).

Objekty mají pořadová čísla (v hranatých závorkách), od 0 výše. Každý objekt musí mít zadány minimálně tři vlastnosti (viz dále), a sice TYP objektu, JMÉNO a jméno FORMULÁŘE, tedy informaci, na kterém objektu tento objekt leží (může ležet buď na formuláři nebo přímo na ploše obrazovky).

Uvedu příklad, podrobněji vysvětlím v dalších kapitolách.

```
objekt[1].typ = OBJEKT_FORM;
objekt[1].jméno = "formulář1";
objekt[1].formulář = "Obrazovka"; // Pozor! začíná velkým písmenem
objekt[1].editace = true;        // true znamená Ano, false Ne
objekt[1].vlevo = 50;   // zatím rozměry jen od oka
objekt[1].nahoře = 50;
objekt[1].šířka = 100;
objekt[1].výška = 100;
```

Nyní se přepněte na globální proměnné (Ctrl + G) a upravte řádek

```
#define POČET_OBJEKTŮ 1
```

na

```
#define POČET_OBJEKTŮ 2   /* máme již 2 objekty: Obrazovka a formulář1 */
```

Když na toto někdy zapomenete, VBalt nebude fungovat nebo bude blbnout!!!

Ještě přidejte řádek

```
#define formulář1 objekt[1]
```

To vám umožní později psát jednoduše např. `formulář1.vlevo = 100;`

Všimněte si ještě řádku

```
#define SOUBOR_EDITACE "*\\temp.$$$"
```

Do tohoto souboru se budou ukládat pomocná data za běhu programu (vysvětlím). Pokud vám toto jméno a cesta nevyhovuje, můžete změnit.

Nyní program spusťte (F9). Objeví se vám obrazovka a formulář. Můžete ho přetahovat (za horní pruh) a roztahovat (za hrany a rohy), jako ve Windows.

Až umístíte formulář tak, jak ho chcete, ťukněte na něj pravým tlačítkem myši. Protože `formulář.editace` je `true`, po stisnutí toho pravého tlačítka se do toho souboru **temp.$$$** uložili aktuální souřadnice objektu. Ukončete program (Ctrl+Break), zapněte manažér (Ctrl+M) a zvolte zobrazení všech souborů (volba 3.). Otevřete **temp.$$$**, zkopírujte to do schránky a můžete to nalepit místo odhadovaných souřadnic do té operace **ů**. A je to. Nyní ještě zrušte editaci (`.editace = false;`) Můžete zkusit přidat ještě tyto řádky:

```
objekt[1].viditelné = true;  /* když není editace, objekty normálně nejsou vidět */
objekt[1].popis = "Formulář č.1";
objekt[1].barva_pozadí = Světlošedá;
objekt[1].myš_dblclick = &_konec; /* po dvojkliku se zavolá funkce _konec,
```

která nastaví globální proměnnou `konec_jádra` na `true`, což je signál pro jádro VBaltu, že se má ukončit. Můžete si samozřejmě vyrobit jinou funkci a do ní napsat `konec_jádra = true;` jenom nesmíte zapomenout v operaci **ů** dát před ni znak **&** */

Zkuste si to zase přeložit (F9). Fajn, už žádný Ctrl+Break. Ještě vám ukážu, jak si tam dáte tlačítko a pak již popíšu detailně všechny objekty, jejich vlastnosti, události a další fígle...

2. 1. Přidáváme tlačítko...

Přepněte se do operace **ů** a přidejte tyto řádky:

```
objekt[2].typ = OBJEKT_TLAČÍTKO;
objekt[2].jméno = "tlač1";
objekt[2].formulář = "formulář1";
objekt[2].editace = true;
objekt[2].vlevo = 0;   // rozměry zase jen od oka
objekt[2].nahoře = 0;
objekt[2].šířka = 30;
objekt[2].výška = 30;
```

Přepněte do Ctrl+G, POČET_OBJEKTŮ 3, přidat řádek

```
#define tlač1 objekt[2]
```

Vždyť už to znáte... F9, umístěte ho hezky (jo mimochodem, veškeré pozice `.vlevo` a `.nahoře` se vztahují relativně k formuláři uvedenému ve vlastnosti `.formulář`). Pravým tlačítkem ťuknout, dvojklikem na formulář to ukončíte, otevřete **temp.$$$** (Trochu teď ve svém výkladu zrychluji, ale tak předpokládám, že od vás, jako programátorů, mohu očekávat jistou dávku inteligence :-)

Nakopírujte souřadnice, zrušte editaci a přidejte do **ů** tyto řádky:

```
objekt[2].viditelné = true;
objekt[2].popis = "PÍP";
objekt[2].barva_pozadí = Žlutá;
objekt[2].myš_click = &píp;
```

Vytvořte novou funkci `píp`, ve stylu:

```
píp
0!
```

a do operace 0 napište `Tón(2000); Čekej(10); VypniTón();`

Tak, a přeložte F9. A už pípáme. Myslím, že by to na úvod stačilo. Běžte se proběhnout, postavit sněhuláka a pod., zítra pokračujem (podíváme se na to trochu víc vědecky, ale budete mít kompletní přehled a už budete vše umět.)

3. Objekty

Nyní již jdeme plně na věc. Opět vás upozorním, že na tom nic není. Ve VBaltu jsem implementoval mnoho objektů, v dalších kapitolkách se podíváme na každý z nich detailně. Nejdříve ale vysvětlím pár pojmů, které je potřeba znát.

3. 1. Typy objektů

Ve VBaltu jsou implementovány následující typy objektů:

`OBJEKT_NIC`	zvláštní typ – tento objekt je ignorován, jako kdyby prostě nebyl
`OBJEKT_FORM`	formulář (chcete-li okno), na něm potom leží další objekty
`OBJEKT_TLAČÍTKO`	tlačítko (může být i s obrázkem :-)
`OBJEKT_TEXT`	okénko pro psaní textů (ve Visual Basicu znám jako TextBox)
`OBJEKT_OBRAZOVKA`	zvláštní typ – objekt **vždy** s indexem 0 (tedy `objekt[0]`), každý program musí mít právě jednu obrazovku. Na ní můžou ležet formuláře nebo i jednotlivé objekty.
`OBJEKT_POSUVNÍK_H`	horizontální posuvník (zleva doprava, vodorovný)
`OBJEKT_POSUVNÍK_V`	vertikální posuvník (shora dolů, svislý)
`OBJEKT_VÝBĚR`	zaškrtávací políčko, může být současně zaškrtnuto více výběrů
`OBJEKT_PŘEPÍNAČ`	zaškrtávací políčko, v dané skupině vždy jen jedno zaškrtnuto
`OBJEKT_ČASOVAČ`	zvláštní typ – není vidět. Po daném intervalu (např. 1 sekunda) zavolá funkci (vznikne událost) `.tik()`, tedy třeba každou sekundu dejme tomu pípne.

3. 2. Vlastnosti objektů

Každý objekt má své vlastnosti, které určují jeho vzhled a způsob chování. Tyto vlastnosti budete právě ve VBaltu při programování nastavovat. Tady je seznam všech vlastností, které se ve VBaltu vyskytují včetně jejich běžných významů. Mějte ale na paměti, že se jedná pouze o přehled. Každý typ objektu používá pouze některé z těchto vlastností a jejich význam může být ještě u každého typu trochu odlišný. Doporučuji tuto kapitolku spíše jen prolétnout a pak se k ní vrátit, až již budete vědět více o jednotlivých objektech.

Pokud chceme vlastnost objektu nastavit, napíšeme

```
objekt[index_objektu].jméno_vlastnosti = hodnota;
```

Já ale osobně doporučuji, že jakmile vytvoříte nový objekt, tak do globálních definic (Ctrl+G) zavedete synonymum, např.

```
#define jméno_objektu objekt[index_objektu]
```

Pak již můžete psát jednoduše např.

```
jméno_objektu.jméno_vlastnosti = hodnota;
```

Nezapomeňte také po každém přidání objektu změnit globální konstantu POČET_OBJEKTŮ. O přidávání objektů pojednám ještě jednou v samostatné kapitole. Já vím, tyto pokyny již píšu podruhé a slibuji vám nyní, že se ještě jednou o nich zmíním. Jsou ale natolik důležité, že to je potřeba. Pokud totiž nezměníte POČET_OBJEKTŮ, program nebude fungovat správně (překladač přitom ale pravděpodobně neohlásí žádnou chybovou zprávu!)

<u>Povinné vlastnosti pro každý objekt:</u>

`int typ`	typ objektu, viz kapitola 3.1. Typy objektů
`string formulář`	jméno objektu, ke kterému objekt přísluší (buď formulář nebo přímo obrazovka. Formulář nemůže ležet v jiném formuláři!)
`string jméno`	jméno objektu (třeba tlačítko1). Ať vám ruka upadne, jestli toto jméno změníte za běhu programu – mimo operaci **ů**, samozřejmě!!! ;-)

Nepovinné vlastnosti objektů:

`char editace`	při návrhu objektů nastavte na `true`, bude vám umožněno přesunovat objekt a měnit jeho rozměry. Pokud ťuknete pravým tlačítkem myši na takovýto objekt, jeho aktuální pozice se uloží do souboru definovaném v globálních definicích (řádek `#define SOUBOR_EDITACE`). Po umístění objektu změňte editaci na `false`. Pokud `.editace` je `true`, pak je objekt vždy vidět (nehledě na vlastnost `.viditelné`) a nevolají se ošetření událostí (tedy např. tlačítko se nedá stisknout a pod.)
`char aktivní`	pod Visual Basicem známé jako `.enabled`, pokud je `false`, neošetřují se události (např. tlačítko se nedá stisknout)
`char viditelné`	viditelnost objektu. Pokud `editace` je `true`, objekt je vždy vidět. U objektu Obrazovka má tato vlastnost speciální význam – obrazovka je samozřejmě vidět vždy, ale jedná se o viditelnost okraje (rámečku kolem Baltazarova prostoru).
`char přesun`	povolení pro formulář, jestli ho smí uživatel přemísťovat myší (drag and drop). Hodnoty `true`, `false`.
`char roztahování`	jako přesun, ale týká se roztahování formuláře
`char ploché`	pro tlačítko, pokud je true, tlačítko je ploché (jako ikonka) a udělá se z něj tlačítko teprve, když na něj najedete myší. (znáte např. z MS Officu 97 a výše)
`int vlevo`	pozice X levého horního rohu objektu (v bodech)
`int nahoře`	pozice Y levého horního rohu objektu
`int šířka`	šířka objektu v bodech
`int výška`	výška objektu v bodech (ale pozor – formulář je ve skutečnosti ještě potom vyšší o titulkový pruh)
`string popis`	nápis na objektu, u objektu Text značí textový obsah okénka
`string obrázek`	jméno souboru s obrázkem (pokud nechceme, tak jen "")
`int barva_pozadí`	barva pozadí objektu (jeho plochy)
`int font_barva`	barva písmenek, případně i třeba okrajů (dle typu objektu)
`int hodnota_min`	pro posuvníky – minimální hodnota (pozice vlevo či nahoře)
`int hodnota_max`	pro posuvníky – maximální hodnota (pozice vpravo či dole)
`int hodnota`	posuvníky – nastavená hodnota výběr, přepínač – hodnota zaškrtnutí (`true`, `false`)
`int šířka_proc`	posuvníky – šířka šoupátka v procentech šířky celého posuvníku (např. 60%, `šířka_proc = 60;`). `šířka_proc` nastavená na –1 znamená automatické přepočítávání šířky šoupátka dle rozsahu hodnot.
`int skupina`	pro přepínače – stejná hodnota skupiny (např. 0) určuje skupinu přepínačů, mezi kterými se přepíná, tedy z této skupiny je zaškrtlý vždy pouze jeden. Označení skupiny –1 používám pro označení, že není součástí žádné skupiny (potřebné pro Výběr a Posuvník)

`char zalamování`	pro objekt Text – `true`, `false`. Zalamování řádku textu, pokud se nevejde do okénka (jinak nutno rolovat)
`char více_řádků`	pro objekt Text – `true`, `false`. Pokud je `false`, jedná se pouze o jednořádkový vstup (např. číslo). Doporučuji pak dát `zalamování` na `false`.
`char edit_text`	pro objekt Text – `true`, `false`. Pokud je false, nedá se text editovat (ve Visual Basicu objekt Label). Pokud je true, text se dá editovat (ve Visual Basicu objekt TextBox)
`char rámeček`	pro objekt Text – `true`, `false`. Viditelnost rámečku okolo objektu.
`int pox, int poy`	pro objekt Text – pozice aktuálního výřezu okénka (pozice 0, 0 značí počátek)
`long pktext`	pozice kurzoru v Textu (označuje pozici znaku ve vlastnosti `.popis`, 1. znak má pozici 0
`int pb1`	pozice začátku bloku (stejně jako `pktext`)
`int pb2`	pozice konce bloku. Pokud `pb1` a `pb2` je -1, blok není.
`long interval`	pro objekt časovač – interval v setinách sekundy mezi jednotlivými událostmi `.tik()` Pokud interval je 0, tak je časovač neaktivní.

3. 3. Události objektů

Události objektu vznikají při nějaké akci související s daným objektem (např. kliknutí na objekt, dvojklik, stisknutí tlačítka myši, puštění tlačítka myši, přejetí myší přes objekt, změna hodnoty atd.) Tyto události pak ve svém programu budete ošetřovat, tedy napíšete funkci, která se má zavolat po vzniku této události. Např. řeknete, že po stisknutí tlačítka se má zavolat funkce, která pípne. Všechny události se volají jen tehdy, je-li objekt aktivní a needitujе se.

Pokud chceme událost ošetřit, naprogramujeme danou funkci a do operace **ů** napíšeme do vlastností objektu:

```
objekt[index_objektu].jméno_události = &jméno_funkce;
```

případně

```
jméno_objektu.jméno_události = &jméno_funkce;
```

<u>Druhy událostí:</u>

`myš_pohyb()`	volá se tehdy, pokud se na objektu pohne myš
`myš_dolů()`	na objektu jsme stiskli levé tlačítko myši
`myš_nahoru()`	na objektu jsme pustili levé tlačítko myši
`myš_click()`	pustili jsme tlačítko na tom samém objektu, na kterém jsme ho stiskli
`myš_dblclick()`	dvojklikli jsme na objekt (maximální čas mezi dvěma kliky dán globální konstantou (Ctrl+G) `DBLCLICK_SETINY`)
`hodnota_změna()`	změna hodnoty u posuvníku, výběru, přepínače, případně změna popisu objektu Text
`tik()`	volá se po časovém intervalu daném vlastností `.interval` v setinách sekundy

<u>Událost stisk klávesy:</u>

Pokud stisknete klávesu, volá se funkce `_obecné_klávesa`, kterou si můžete editovat. V proměnné `gKlávesa` je stisklá klávesa a k dispozici jsou i `shifty` – viz nápověda přímo u funkce (hned pod její hlavičkou).

3. 4. Implicitní nastavení vlastností a událostí objektů

Pokud nenastavíte jinak, pro každý objekt jsou přednastaveny tyto hodnoty:

```
.typ = OBJEKT_NIC;
.formulář = "";
.jméno = "";

.editace = false;
.aktivní = true;
.viditelné = false;
.přesun = true;
.roztahování = true;

.ploché = false;
.popis = "";
.obrázek = "";

.hodnota = 0;
.skupina = -1;
.šířka_proc = -1;

.zalamování = true;
.více_řádků = true;
.edit_text = false;
.rámeček = true;
.pox = 0;
.poy = 0;
.pktext = 0;
.pb1 = -1;
.pb2 = -1;

.interval = 0;

.myš_pohyb = NULL;      // NULL znamená žádné ošetření události
.myš_dolů = NULL;
.myš_nahoru = NULL;
.myš_click = NULL;
.myš_dblclick = NULL;
.hodnota_změna = NULL;
.tik = NULL;
```

3. 5. Přidávání nových objektů

Z adresáře VB_VZOR si zkopírujte do svého vlastního adresáře všechny 3 soubory a přejmenujte je na nový název (například POKUS).

V Baltazarovi otevřete soubor např. POKUS.SGP, v něm budete normálně programovat. Pokud byste ale tento program spustili, nemá zatím nikde naprogramováno, aby se ukončil. Jediné řešení je tedy ukončit pomocí Ctrl + Break.

Přepněte se na funkci __Jádro (všimněte si, že všechny vnitřní funkce VBaltu začínají podtržítkem). Přepněte se na operaci **ů** (hned na začátku skoro úplně vlevo) a editujte ji. Každý program musí obsahovat minimálně jeden objekt, a sice Obrazovka (`objekt[0]`).

Objekty mají pořadová čísla (v hranatých závorkách), od 0 výše. Každý objekt musí mít zadány minimálně tři vlastnosti, a sice TYP objektu, JMÉNO a jméno FORMULÁŘE, tedy informaci, na kterém objektu tento objekt leží (může ležet buď na formuláři nebo přímo na ploše obrazovky).

Pak se přepněte do globálních proměnných (Ctrl+G), změňte POČET_OBJEKTŮ na aktuální počet a na závěr přidejte řádek ve stylu:

```
#define TlačítkoUlož objekt[12]
```

To vám umožní později psát ve funkcích (ošetření událostí) např.

```
TlačítkoUlož.viditelné = false;
```

Do operace **ů** vložte např. tyto řádky:

```
objekt[12].typ = OBJEKT_TLAČÍTKO;
objekt[12].jméno = "TlačítkoUlož";
objekt[12].formulář = "formulář1";
objekt[12].editace = true;
objekt[12].vlevo = 50;   // zatím rozměry jen od oka
objekt[12].nahoře = 50;
objekt[12].šířka = 100;
objekt[12].výška = 100;
```

Po přeložení programu (F9) můžete upravit pozici a rozměry objektu (díky zapnutí editace). Až budete s pozicí spokojeni, ťukněte na objekt pravým tlačítkem a do souboru daného v globálních definicích (implicitně "*\\temp.$$$") se uloží aktuální rozměry.

Ukončete program, otevřete v manažeru **temp.$$$**, rozměry zkopírujte do schránky a upravte je pak v operaci **ů**. Nyní zrušte editaci, objekt máte připraven. Nyní můžete nastavovat další vlastnosti, viz další kapitoly.

Ošetřování událostí:

Přiřaďte adresu funkce do dané vlastnosti, např:

```
TlačítkoUlož.myš_click = &PÍPNI;   // po stisknutí TlačítkoUlož zavolá funkci
                                   // PÍPNI
```

3. 5. 1. Objekt NIC

Předpokládejme, že zatím máme nadefinován pouze `objekt[0]` – objekt Obrazovka.

```
objekt[1].typ = OBJEKT_NIC;
```

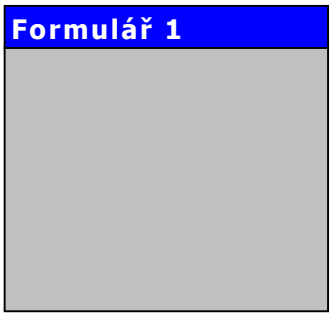
Tento objekt je naprosto ignorován, další vlastnosti také.

1. 5. 2. Objekt OBRAZOVKA

```
objekt[0].typ = OBJEKT_OBRAZOVKA;
objekt[0].formulář = "Obrazovka";
objekt[0].jméno = "Obrazovka";
```

Tento objekt je už ode mě přednastaven. Jediné, co na něm můžete měnit, je vlastnost `viditelné`. Ta u obrazovky určuje viditelnost okrajů Baltazarova prostoru.

Na obrazovce mohou ležet libovolné objekty.



3. 5. 3. Objekt FORM

```
objekt[1].typ = OBJEKT_FORM;
objekt[1].formulář = "Obrazovka";
objekt[1].jméno = "Formulář1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, přesun, roztahování, vlevo, nahoře, výška, šířka, popis, obrázek, barva_pozadí, font_barva`

Objekt formulář je zvláštní objekt – ostatní typy objektů mohou ležet buď přímo na obrazovce, nebo na jednom z formulářů. S tímto formulářem pak sdílí svoje vlastnosti.

Pozice těchto objektů se vztahuje relativně k pozici formuláře. Pokud formulář není vidět, ani tyto objekty v něm nejsou vidět a totéž platí pro vlastnost aktivní. Pokud je formulář menší, než je pozice objektů v něm, jsou vykresleny jen z části, nebo vůbec (je pak potřeba formulář roztáhnout).

Formuláře lze dále přepínat do výškových hladin (když na něj ťuknete, přepne se úplně nahoru).

Formulář může ležet vždy jen na Obrazovce.

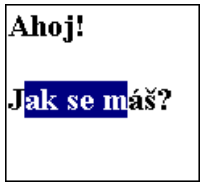
3. 5. 4. Objekt TLAČÍTKO

```
objekt[1].typ = OBJEKT_TLAČÍTKO;
objekt[1].formulář = "Formulář1"; // například
objekt[1].jméno = "tlač1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, ploché, vlevo, nahoře, výška, šířka, popis, obrázek, barva_pozadí, font_barva`

Tlačítko může být ploché – jako ikonka, tlačítko se z něj udělá teprve tehdy, až na něj najedeme myší.

3. 5. 5. Objekt TEXT

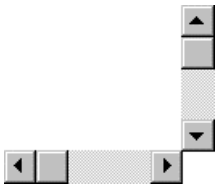



```
objekt[1].typ = OBJEKT_TEXT;
objekt[1].formulář = "Formulář1"; // například
objekt[1].jméno = "text1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, vlevo, nahoře, výška, šířka, popis, barva_pozadí, font_barva, zalamování, edit_text, rámeček, pox, poy, pktext, pb1, pb2`

Obsah tohoto objektu je uložen ve vlastnosti `.popis`. Pokud je `.edit_text true`, pak můžeme editovat text z klávesnice (nejdříve je potřeba na text kliknout myší). Funguje pohyb kurzoru šipkami, označování bloku Shift + šipky, kopírování do schránky (Ctrl+C, Ctrl+Insert), vyjmutí do schránky (Ctrl+X), vložení ze schránky (Ctrl+V, Shift+Insert), klávesy (shift+) PageDown, PageUp, Home, End, Ctrl+Home, Ctrl+End, označení celého textu Ctrl+A.

3. 5. 6. Objekt POSUVNÍK

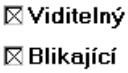


```
objekt[1].typ = OBJEKT_POSUVNÍK_H; // nebo _V
objekt[1].formulář = "Formulář1"; // například
objekt[1].jméno = "posuvka1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, ploché, vlevo, nahoře, výška, šířka, barva_pozadí, font_barva, hodnota_min, hodnota_max, hodnota, šířka_proc`

Co dodat? Ani nevíte, kolik práce mi to dalo tento objekt naprogramovat... ;-)

3. 5. 7. Objekt VÝBĚR

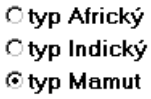



```
objekt[1].typ = OBJEKT_VÝBĚR;
objekt[1].formulář = "Formulář1"; // například
objekt[1].jméno = "vyb1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, vlevo, nahoře, výška, šířka, popis, barva_pozadí, font_barva, hodnota`

Hodnota zaškrtnutí je dána vlastností `hodnota`. Můžete mít zaškrtnuto výběrů, kolik chcete (najednou).

3. 5. 8. Objekt PŘEPÍNAČ




```
objekt[1].typ = OBJEKT_PŘEPÍNAČ;
objekt[1].formulář = "Formulář1"; // například
objekt[1].jméno = "přep1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`editace, aktivní, viditelné, vlevo, nahoře, výška, šířka, popis, barva_pozadí, font_barva, hodnota, skupina`

Hodnota zaškrtnutí je dána vlastností `.hodnota`. Můžete mít zaškrtnut vždy pouze jeden přepínač ze stejné skupiny přepínačů (vlastnost `.skupina`).

3. 5. 9. Objekt ČASOVAČ

```
objekt[1].typ = OBJEKT_ČASOVAČ;
objekt[1].formulář = "Obrazovka";
objekt[1].jméno = "čas1"; // například
```

Pro tento objekt se vztahují dále tyto vlastnosti:
`aktivní, interval`

Pokud není `interval` 0, tak po tomto intervalu v setinách sekundy vznikne událost `.tik()`.

3. 6. Přehled objektů

		Objekt								
vlastnosti	typ	nic	obrazovka	form	tlačítko	text	posuvník	výběr	přepínač	časovač
	jméno		x	x	x	x	x	x	x	x
	formulář		x	x	x	x	x	x	x	x
	editace			x	x	x	x	x	x	
	aktivní			x	x	x	x	x	x	x
	viditelné		x	x	x	x	x	x	x	
	přesun			x						
	roztahování			x						
	ploché				x		x			
	vlevo		x	x	x	x	x	x	x	
	nahoře		x	x	x	x	x	x	x	
	šířka		x	x	x	x	x	x	x	
	výška		x	x	x	x	x	x	x	
	popis		x	x	x	x		x	x	
	obrázek			x	x					
	barva_pozadí		x	x	x	x	x	x	x	
	font_barva		x	x	x	x	x	x	x	
	hodnota_min						x			
	hodnota_max						x			
	hodnota						x	x	x	
	šířka_proc						x			
	skupina								x	
	zalamování					x				
	více_řádků					x				
	edit_text					x				
	rámeček					x				
	pox					x				
	poy					x				
	pktext					x				
	pb1					x				
	pb2					x				
	interval									x
události	myš_pohyb		x	x	x	x	x	x	x	
	myš_dolů		x	x	x	x	x	x	x	
	myš_nahoru		x	x	x	x	x	x	x	
	myš_click		x	x	x	x	x	x	x	
	myš_dblclick		x	x	x	x	x	x	x	
	hodnota_změna					x	x	x	x	
	tik									x

4. Pár rad na závěr...

Věřím, že se vám můj program líbí. Doufám, že vám hodně pomůže a ušetří spoustu práce.

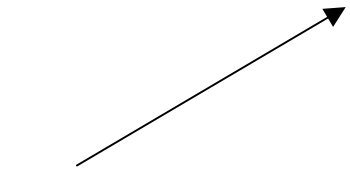
Jo, ještě o jedné věci se zmíním. SGP Baltazar 5.0 neposkytuje možnosti pro ochranu přístupových práv k jednotlivým proměnným (nejedná se o objektově orientovaný jazyk), tudíž je vám umožněno změnit jakoukoliv systémovou proměnnou. Pokud ale změníte něco, co nesmíte, může to mít pro následující běh programu katastrofální účinky, dejte si tedy na to pozor. Z globálních proměnných a vlastností objektů měňte jen ty, o kterých jsem psal. Ostatních si nevšímejte.

Nezaručuji, že je můj program bez chyb! Dělal jsem ho ve velkém spěchu, a i kdybych na něj měl mnoho času, určitě by nebyl bezchybný. Proto, jestli narazíte na nějaké divné chování, informujte mne, prosím.

Je třeba zaslat mi všechny soubory vašeho programu a snažte se, prosím, specifikovat divné chování tak, abych si ho uměl přesně podle návodu vymodelovat (tedy žádné: no, něco jsem nějak stisknul a ono to tam potom nakreslilo bílou čáru). Chci přesný popis úplně od začátku spuštění programu až k vlastní chybě. Každému nálezci chyby předem děkuji.

Pokud by vás zajímalo, jak se takový program, jako je VBalt 5.0, dělá, tak se můžete podívat do adresáře s dokumentací, jsou tam naskenované nějaké papíry, na které jsem si po celou dobu programování dělal poznámky. Je to docela zajímavé...

Tomáš Bořil I.
borilt@email.cz



to jsem já

A. Příloha: používání nástroje vbBMPfik

Nástroj vbBMPfik slouží k dělání výřezů z velkých `.BMP` obrázků. SGP Paint vám totiž umožňuje uložit pouze celou obrazovku. Když ale potřebujete dát na objekt obrázek, budete potřebovat jen výřez této obrazovky (ten, kam jste si daný malý obrázek nakreslili). Postup vyříznutí takového obrázku je následující:

Např. chcete si udělat obrázek na tlačítko – nakreslete ten obrázek v SGP Paintu, uložte celou obrazovku jako `.BMP`, spusťte vbBMPfik, načtěte toto `.BMP`, označte výřez s obrázkem na tlačítko, uložte výřez. Toť vše (a nezapomeňte vbBMPfik ukončit ;-).

Takto vyrobené obrázky již můžete používat do programů vytvořených ve VBaltu. Asi nejlepší je nakopírovat je do stejného adresáře, ve kterém máte svůj program a pak se můžete na tyto obrázky odkazovat jednoduše, například:

```
tlačítko1.obrázek = "*\\nacti.bmp";
```

B. Příloha: ASCII tabulka ;-)

abcdefghijklmnopqrstuvwxyz