

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Evidence hodnocení studentů čárovými kódy

Plzeň, 2006

Petr Muchna

Abstract

Evidence of students appreciation by barcodes

A purpose of this bachelor thesis is to develop student agenda based on identification through barcodes. Program is implemented in .NET framework 2.0. Barcode printing is realized through barcode font. The system uses CODE39 barcode standard. System cooperates with university student agenda – STAG, based on the Oracle database system.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 27.8.2006

.....
Petr Muchna

Obsah

Úvod	6
1 Čárové kódy a automatická identifikace	7
1.1 Úvod do problematiky	7
1.2 Rozdělení čárových kódů	7
1.3 Parametry čárových kódů	9
1.4 Standard CODE39	10
1.4.1 Použití	10
1.4.2 Parametry	10
1.4.3 Kontrolní znak	11
1.4.4 Code39 Full ASCII	11
2 Ostatní standardy čárových kódů	11
2.1 Kódy 2 z 5	11
2.2 Code 128	12
2.3 Code 93	12
2.4 Codabar	12
2.5 EAN	12
3 Zařízení pro čtení čárového kódu	14
3.1 Části čtecího zařízení	14
3.2 Typy čtecích zařízení a podporovaná rozhraní	14
3.3 Referenční čtečka	16
4 Tisk čárových kódů	17
5 Návrh řešení	18
5.1 Navázání aplikace na systém STAG	18
5.2 Volba řetězce pro zakódování	19
5.3 Zabezpečení čárového kódu proti zneužití	20
5.4 Provázání kódu s testem a tisk čárových kódů	20
5.5 Způsob evidence hodnocení	21
5.6 Výstup a propagace výsledků hodnocení na univerzitním portálu	21
6 Realizace aplikace	22
6.1 Připojení k databázi Oracle	22
6.2 Tisk čárových kódů v aplikaci	22
6.2.1 GDI+	23
6.2.2 Font čárového kódu	23
6.2.3 Postup při vytvoření řetězce pro tisk fontem čárového kódu	23
6.2.4 Řezané štítky	24
6.3 Lokální uložení	25
7 Závěr	26
8 Seznam zkratk	27

Literatura	28
A Příloha – Uživatelská příručka	29
A.1 Popis uživatelského rozhraní programu	29
A.2 Správa termínů	29
A.2.1 Přiřazení testu	30
A.2.2 Tisk čárových kódů	30
A.3 Vstup hodnocení	31
A.4 Export výsledků	31
A.5 Připojení k databázi studijní agendy STAG	32
A.6 Instalace aplikace	32
A.7 Nastavení referenční čtečky	33
A.8 Použití aplikace	33

Úvod

V dnešní době stále rostou požadavky na rychlost zpracování dat. Hledají se různá řešení, jedním z nich je automatická identifikace a její významné odvětví - čárové kódy. Cílem této práce je navrhnout aplikaci, která umožní evidovat hodnocení studentů čárovými kódy a tím usnadní a urychlí proces organizace a vyhodnocování písemných zkoušek. Aplikace musí spolupracovat se systémem univerzitní agendy STAG a importovat z něho data potřebná pro tisk sestav čárových kódů. Vlastní systém hodnocení umožní exportovat výsledky tak, aby byly dále snadno zpracovatelné a v podobě, která umožní jejich propagaci na internetu i v rámci univerzitního portálu.

Práce je členěna na teoretickou část, která zahrnuje první 3 kapitoly, a část praktickou popsanou v posledních 2 kapitolách. V 1. kapitole se zabývám parametry čárových kódů a detailně popisuji standard CODE39. V další kapitole je uveden stručný přehled nejpoužívanějších standardů a jejich základní vlastnosti. Kapitoly 3 a 4 se zabývají tiskem čárových kódů a zařízeními umožňujícími čtení čárových kódů.

V následující kapitole popisuji návrh řešení a specifikuji konkrétní záznamy, které budou importovány ze STAGu. Zabývám se volbou standardu čárového kódu, strukturou identifikátoru pro zakódování a samotným způsobem kódování. Též rozvíjím problematiku přiřazení kódu ke konkrétnímu testu a studentovi, rovněž i otázku bezpečnosti a zneužití čárového kódu pro změnu identity studenta v rámci zkoušky.

V poslední kapitole popisuji hlavní problémy, které bylo nutné vyřešit při realizaci aplikace. Jednalo se nejen o volbu programovacího jazyka, ale i prostředí pro vývoj aplikace a způsob připojení k Oracle databázi univerzitní agendy. V neposlední řadě to byl i tisk sestav čárových kódů prostřednictvím grafického rozhraní GDI+ a rovněž způsob ukládání dat a hodnocení. V jednotlivých podkapitolách jsou uváděny popisy tříd aplikace, kde je daná problematika řešena.

1 Čárové kódy a automatická identifikace

1.1 Úvod do problematiky

Problematika čárových kódů spadá do oblasti automatické identifikace (dále jen AI). Ta se masově rozšířila s rozvojem výpočetní techniky a dnešní rozšíření počítačů další vývoj jen podporuje. První patent na kruhový čárový kód přihlásil Američan N. J. Woodland roku 1949. Jednalo se o patent, který měl urychlit odbavení zákazníka u pokladen supermarketů. Od té doby můžeme pozorovat velký rozmach tohoto odvětví. V roce 1973 byla přijata symbolika UPC (Universal Product Code) na jednotné označování výrobků a čárový kód se prosadil definitivně. V Evropě (a také v tehdejší Československu) se čárový kód objevuje v roce 1976 se zavedením symboliky EAN (European Article Numbering) a s ní spojených závazných doporučení a norem.

Při zpracování čárových kódů jsou kladeny vysoké nároky na zařízení sloužící ke tvorbě, přenosu a sběru dat. Prvky AI musí eliminovat rizika lidských chyb při zpracování a zajistit vysokou spolehlivost v nejnáročnějších podmínkách. Použití čárových kódů, jakožto nejznámějšího způsobu AI, je jednoduché a cenově nenáročné. Vyznačuje se všeobecným použitím s možností aplikace na téměř libovolný materiál (plast, papír, textil, kov, keramika atd.).

Ke čtení čárových kódů se používá široké spektrum snímacích zařízení (např. snímací pera, CCD scannery, laserové scannery - ruční, přenosná, automatická atd.). Čtecí zařízení musí splňovat přísné nároky jako je rozsah provozních teplot, nízká spotřeba energie, ergonomie, mobilita, vysoká přesnost, rychlost a spolehlivost snímání. Čtení může být plně automatizováno nebo poloautomatizováno. Vždy záleží na nárocích, jaké jsou na celý systém kladeny. Spolehlivost sběru dat pomocí čtecích zařízení je mnohonásobně větší než ruční zadávání. Čtecí zařízení vždy podporují širokou paletu všech základních a nejpoužívanějších čárových kódů.

1.2 Rozdělení čárových kódů

V dnešní době se lze setkat s více než 200 standardy čárových kódů. Některé jsou všeobecně rozšířené, jiné jsou vhodné pouze pro určité průmyslové odvětví nebo se používají jen v určité lokalitě. Výběr čárového kódu je podmíněn řadou faktorů, především používanou technologií (tisk, materiál, ...) a požadavky na kódovaná data. Čárové kódy lze rozdělit do několika základních kategorií.

1. Podle znaků, které je schopen zakódovat:

- Kódy numerické
- Kódy numerické se speciálními znaky
- Kódy alfanumerické
- Kódy úplné alfanumerické (Full-ASCII)

2. Podle délky čárového kódu:

- Kódy s pevnou délkou

- Kódy s variabilní délkou

3. Podle oblasti použití:

- Kódy používané v průmyslu
- Kódy používané v obchodech
- Speciální kódy pro armádu, zdravotnictví . . .

Vývoj ukázal, že optimální čárový kód by měl mít tyto parametry:

- Rozsáhlý soubor znaků
- Automatická kontrola načtení (např. pomocí kontrolního znaku)
- Konstantní šířka kódů znaku
- Stejný počet čar a mezer pro všechny znaky
- Jednoduchá struktura kódu (dvě různé šířky prvků kódu - široká a úzká čára/mezera)
- Možnost libovolné rychlosti snímání (např. při ručním sběru dat)
- Tolerance při tištění kódu a z toho plynoucí tolerance při dekodování
- Vysoká hustota kódu (minimální pokrytý prostor)

Následující tabulka obsahuje stručný přehled nejpoužívanějších standardů a jejich charakteristiku :

Název	Počet znaků	Typ	Délka	Použití
UPC A	10	1	12	obchod (USA, Kanada)
UPC E	10	1	8	obchod (USA, Kanada)
EAN 8	10	1	8	obchod (Evropa)
EAN 13	10	1	13	obchod (Evropa)
Code 2/5	10	1	variabilní	technika
Codabar	16	1 + S	variabilní	fotoprůmysl, knihovny
Code 39	43	1 + S + A	variabilní	všeobecné použití, farmacie
Code 93	43	1 + S + A	variabilní	farmacie, elektrotechnika
Code 128	128	1 + S + A + a	variabilní	technika, farmacie, medicína

Vysvětlivky: 1 - numerický, S - speciální znaky, A - velká písmena, a - malá písmena

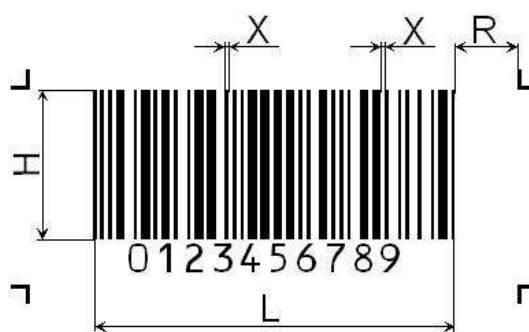
Tabulka 1: Nejpoužívanější standardy čárových kódů

Volba čárového kódu tedy hlavně závisí na tom, které znaky jím potřebujeme vyjádřit. Nejrozšířenějšími čárovými kódy jsou Code 39, UPC a EAN. Nejnovějšími typy čárových kódů jsou dvojrozměrné, které zobrazují informace ve více řádcích a dosahují tak vysoké hustoty dat. Představitelem tohoto standardu je např. PDF417, další informace o těchto kódech lze nalézt v [1]

1.3 Parametry čárových kódů

Čárový kód je tvořen posloupností čar a mezer. Nosičem informace jsou jak čáry, tak mezery, které mohou být různě široké. Předpis řazení čar a mezer a jejich šířky je specifický pro každý standard kódu a je to základní ukazatel jednotlivých druhů kódu. Z tohoto důvodu se liší množství informace, kterou je kód schopen zaznamenat na jednotce plochy. Při čtení kódu čtecí zařízení generuje elektrické impulsy, které odpovídají sekvenci tmavých čar a světlých mezer. Po rozpoznání sekvence dostaneme na výstupu řetězec odpovídající zakódované informaci.

Každý znak v kódu je zakódován podle kódovací tabulky, která určuje posloupnost čar a mezer. Kromě kódovaných znaků je každý čárový kód uvozen „start znakem“ a ukončen tzv. „stop znakem“. Start a stop znak slouží pro rozpoznání typu kódu čtecím zařízením. V některých kódech (např. EAN 8 a EAN 13) se vyskytují také znaky dělicí, které rozdělují řetězec do více částí. Každý kód musí být ohraničen světlým pásmem neboli klidovou zónou. Toto pásmo nesmí obsahovat žádné grafické ani textové symboly a slouží čtecím zařízením pro lepší nalezení začátku a konce kódu. Každý kód, pokud není v jeho specifikaci uvedeno jinak, by měl splňovat následující podmínky:



Obrázek 1: Parametry čárového kódu

- X** šířka modulu, specifikuje nejužší element kódu (čára i mezera)
- R** světlé pásmo, rovno desetinásobku šířky modulu, ne však méně než 2,5mm
- L** délka kódu bez světlého pásma
- H** výška čárového kódu

Velikost kódu závisí na hodnotě modulu X . Čím je modul X menší, tím jsou nároky na čtecí zařízení a přesnost tisku čárového kódu větší. Podle hodnoty modulu definujeme parametr hustota kódu, který rozdělujeme do několika skupin.

Chyby při čtení kódu se vyskytují velice zřídka. Téměř jediný důvod pro chybné přečtení kódu bývá jeho mechanické poškození (roztržení, odření, nekvalitní tisk, špína apod.). Ve většině případů čtecí zařízení poškozený kód vůbec nerozpozná, ale v případech, kdy sejmutá data i z poškozeného kódu odpovídají nějaké sekvenci čar a mezer z kódovací tabulky, je rozpoznán chybný řetězec. Řešením pro detekci této chyby je přidání kontrolního znaku, který se vypočítá ze všech předchozích znaků. Kontrolní znak je již pevnou součástí některých čárových kódů (např. EAN 8 a EAN 13) a u jiných kódů je tento znak volitelný (např. Code 39).

Hustota	Šířka modulu v mm
Velmi vysoká hustota	$X < 0,19$
Vysoká hustota	$0,19 < X < 0,24$
Střední hustota	$0,24 < X < 0,30$
Nízká hustota	$0,30 < X < 0,50$
Velmi nízká hustota	$X > 0,50$

Tabulka 2: Hustota čárových kódu v závislosti na šířce modulu

1.4 Standard CODE39

1.4.1 Použití

Kód Code 39 byl vyvinut firmou Intermec v roce 1974 a byl prvním alfanumerickým čárovým kódem, díky čemuž se rychle a úspěšně rozšířil. V roce 1980 byl přijat za standard v americké armádě a roku 1981 začal být používán pro označování obchodních kontejnerů. Kód je tzv. bidirekcionální, je možné ho snímat odpředu i odzadu. Kód je zvláště vhodný pro aplikace vyžadující identifikaci v rozsahu celého alfanumerického souboru znaků, ale využívá se i v aplikacích čistě numerických pro svoji spolehlivost a rozšířenost. Kód můžeme najít ve skladovém hospodářství, řízení výroby a obchodě.

1.4.2 Parametry

Kód podporuje numerické znaky 0 až 9, znaky velké abecedy A až Z a speciální znaky "\$", "/", "+", ".", "%". Start a stop znakem je znak "*". Struktura kódu je pro číslice 0 až 9 shodná s kódem 2/5 viz. dále. Každý diskretní kód (jeden znak převedený na čárový kód) se skládá z pěti čar a čtyř mezer. Tři elementy jsou široké a zbývající úzké. Odtud plyne název kódu, kdy 3 elementy z 9 jsou široké. Nositeli informace jsou jak čáry, tak mezery.

Maximální délka je závislá na parametrech čtecího zařízení, avšak nejčastěji je stanovena na 32 znaků. Existují ale i zařízení podporující délku 48 až 64 znaků. Nejmenší možná šířka modulu X je 0,19 mm pro vysokou hustotu, 0,3 mm pro střední hustotu a 0,53 mm pro nízkou hustotu. Poměr šířky širokého elementu k elementu úzkému je 3:1, přičemž se tato hodnota může pohybovat v rozmezí 2:1 až 3:1. Pro šířku modulu X menší než 0,5 mm platí minimální poměr 2,25:1. Mezi dvěma diskretními kódy v Code 39 je meziznaková mezera, která odpovídá šířce modulu X . Je povoleno, aby tato mezera byla až třikrát větší. Okrajové světlé pásmo má mít šířku 20 – 30 modulů X , minimálně 10 modulů. Pro spolehlivé ruční snímání kódu se doporučuje výška kódu alespoň 6 mm nebo 15% z délky celého kódu.

Je připuštěna tolerance rozměrů, která se dá získat pomocí vzorce s dosazenými hodnotami šířky modulu X a poměru šířek elementů P :

$$T = \pm \left(\frac{4}{27} \cdot \left(P - \frac{2}{3} \right) \cdot X \right)$$

Délku kódu můžeme vypočítat podle vzorce:

$$L = 2 \cdot R + N \cdot (3 \cdot P \cdot X) + 2 \cdot (3 \cdot P \cdot X + 6 \cdot X) + (N + 1) \cdot M$$

L délka kódu včetně světlého pásma [mm]

N počet kódovaných znaků

X šířka modulu X [mm]

P poměr širokých elementů k úzkým

M šířka mezery mezi diskrétními kódy [mm]

R šířka světlého pásma

1.4.3 Kontrolní znak

Code 39 je kód s dokonalou samokontrolní strukturou a ve většině běžných aplikací se kontrolní znak nepoužívá. Při splnění parametrů a kvalitním tisku je možné očekávat chybu jednou za 70 milionů čtení. Kontrolní číslice je vyžadována např. ve zdravotnictví. Výpočet kontrolní číslice se provádí systémem MODULO 43. Každý znak má v kódovací tabulce svůj číselný ekvivalent (0 - 43). Nejdříve se tyto ekvivalenty sečtou a poté vydělí MODULO 43. Znak, na který převedeme výslednou hodnotu pomocí kódovací tabulky, je hledaný kontrolní znak.

1.4.4 Code39 Full ASCII

Tento kód je rozšířením kódu Code 39 na úplnou ASCII tabulku. Plně vychází z kódovací tabulky pro Code 39. Znaky, které Code 39 nepodporuje se v Code 39 Full ASCII zapisují kombinací jednoho ze speciálních znaků ("\$", "/", "%", "+") se znakem velké abecedy. K rozpoznání tohoto kódu se používá rozšířený Start znak "\$*" a Stop znak "*-\$". Kontrolní znak se počítá stejným způsobem jako u Code 39.

2 Ostatní standardy čárových kódů

2.1 Kódy 2 z 5

Kódy skupiny 2/5 (Two of Five) byly jedněmi z prvních, které byly široce přijaty k používání (rok 1968). Tyto kódy mají mnoho variant. Jedná se o kódy numerické s variabilní délkou. Jsou tvořeny Start a Stop znaky a číslicemi 0 až 9. Další společné vlastnosti jsou (až na výjimky):

- každý diskrétní kód obsahuje tři úzké a dva široké elementy (podle názvu 2 z 5)
- mezery mezi čarami nenesou žádnou informaci, tedy významové jsou pouze čáry
- kontrolní číslice není povinná a počítá se MODULO 10
- nízká informační hustota na jednotku délky

2.2 Code 128

Code 128 je alfanumerický kód s variabilní délkou a je tvořen 128 ASCII znaky, 4 speciálními, 4 řídicími, 3 Start znaky a 1 Stop znakem. Code 128 má celkem tři sady znaků označované jako kód A, kód B a kód C. Sada A obsahuje znaky velké abecedy, numerické znaky, řídicí a speciální znaky. Sada B obsahuje stejné znaky jako sada A a navíc znaky malé abecedy. Sada C obsahuje dvojice numerických znaků od 00 do 99, řídicí a speciální znaky. Řídicí a speciální znaky u jednotlivých sad slouží k přepínání mezi sadami znaků v rámci jednoho čárového kódu. Diskrétní kód se skládá ze tří čar a tří mezer. Kontrolní znak se počítá MODULO 103.

2.3 Code 93

Code 39 má jednu nevýhodu a to tu, že je velice plošně náročný. Pokud máme delší řetězec, vede to k příliš dlouhému čárovému kódu, který je náročnější na umístování a snímání. Code 93 tento problém vyřešil, protože má vysokou plošnou hustotu. Stalo se tak odstraněním meziznakové mezery a použitím šesti elementů namísto devíti. Znaková sada je shodná s kódem Code 39. Nevýhodou kódu je, že není samokontrolní. Z toho důvodu se používají dva kontrolní znaky počítané MODULO 47. Code 93 existuje i ve verzi Full ASCII, jejíž princip je shodný s Code 39.

2.4 Codabar

Čárové kódy Codabar původně sloužily pro maloobchod a vycházelo se z nich při návrhu pozdějších dokonalejších kódů. Jedná se o numerický kód s variabilní délkou. Zahrnuto je rovněž šest speciálních znaků a čtyři různé Start a Stop znaky, které ovšem mají stejný účel. Diskrétní kódy jsou odděleny mezerou. Kontrolní znak je počítán MODULO 16, ale vzhledem k tomu, že kód je samokontrolní s vysokou spolehlivostí, se nepoužívá.

Verze ABC-Codabar byla přijata jako standard k označování krevních konzerv a tím došlo k jeho rozšíření ve zdravotnictví. Jedinou nutnou podmínkou tohoto kódu je, že musí být tvořen Start a Stop znakem a mezi nimi minimálně třemi kódovanými znaky. Je možno použít rovněž dvojitý kód ABC-Codabar, který se skládá ze dvou samostatných čárových kódů umístěných vedle sebe. Výstupem je jeden řetězec složený z obou kódů.

2.5 EAN

V roce 1973 byl v USA za standard přijat čárový kód UPC (Universal Produkt Code). Díky velkému zájmu ze zahraničí z něj byl roku 1977 vyvinut kód EAN (European Article Numbering). Kódy UPC a EAN jsou tedy navzájem kompatibilní, přičemž dnes je UPC podmnožinou EAN. Používání kódu EAN řídí mezinárodní organizace IANA EAN (International Article Numbering Association EAN) se sídlem v Belgii, jejímiž členy jsou jednotlivé národní organizace. V ČR a SR je jí CSS EAN. Čárový kód EAN je normalizován samozřejmě i v České republice (ČSN 770060, ČSN 770061, ČSN 770062, ČSN 770064, ČSN 9771101). Stejně jako u CODE 39 i u kódu EAN dochází k postupnému přejímání evropských norem.

EAN 13 i EAN 8 je numerický kód s pevnou délkou (13, popř. 8 číslic). Skládá se z okrajových znaků, dělicího znaku a číslic 0 až 9. Jsou definovány tři kódovací tabulky

(A pro lichou paritu a B a C pro sudou paritu). O použití kódovací tabulky pro kód EAN 13 rozhoduje znak na 13. pozici zprava. U kódu EAN 8 se vždy použije tabulka A pro první čtyři číslice a tabulka C pro zbývající. Výsledný čárový kód EAN se skládá ze dvou částí oddělených dělicím znakem. Diskrétní kód má délku sedmi modulů a skládá se ze dvou čar a dvou mezer se šířkou 1 až 3 moduly.

Kontrolní znak je u kódu EAN povinný. Rozměry kódu a tolerance je určena složitým systémem tabulek a výpočtů. Specifikace předepisuje i přesné umístění, velikost a typ písma pro kódovaný řetězec umístěný pod kódem.

Kódy EAN mohou obsahovat i tzv. dodatkové kódy (EAN with ADD ON), jimiž se kódují různé dodatkové informace. Můžeme je nalézt například na časopisech (vyjádření pořadového čísla časopisu v roce), knihách atd. Dodatkový kód se používá dvoumístný nebo pětímístný a kóduje se podle tabulek A a B. Začíná okrajovým znakem, následuje první zakódovaný znak, dělicí znak, druhý zakódovaný znak, dělicí znak, . . . Právý okrajový znak chybí a kontrolní znak se nevyužívá. Při čtení kódu EAN s dodatkovým kódem na výstupu dostaneme jeden řetězec. Čárových kódů EAN může využívat kdokoliv, kdo se zapojí do systému EAN a uzavře smlouvu s organizací koordinující využívání tohoto systému.

3 Zařízení pro čtení čárového kódu

Pro přenesení informací zapsaných čárovým kódem pro další zpracování, ať už s pomocí počítače nebo lidské síly, slouží snímače čárového kódu, neboli čtecí zařízení. Po úspěšném dekodování dostaneme z čárového kódu původní řetězec.

3.1 Části čtecího zařízení

Prvním elementem čtecího zařízení, který se dostává do "přímého" kontaktu s čárovým kódem je vstupní část (modul). Na něj navazuje modul pro zpracování signálu a zařízení pro logické vyhodnocení elektrických signálů. Poslední částí je modul pro výstup zpracovaných informací. Pro naše potřeby bude postačovat, když čtecím zařízením budeme nazývat první dva moduly a zbylé dva pojmenujeme dekodér. Dekodér slouží pro dekodování a výstup informací, které mu dodá čtecí zařízení.

Kombinací různých druhů čtecích zařízení (scanner, pero atd.) a dekodérů (několik druhů rozhraní) dostaneme mnoho řešení pro čtení čárových kódů. Z tohoto hlediska můžeme čtecí zařízení rozdělit na kompaktní (včetně dekodéru) a nekompaktní. Kompaktní zařízení se používá hlavně pro mobilní sběr dat a vyžadují tedy nízkou hmotnost, vysokou ergonomii a energetickou nenáročnost. Výhodou nekompaktních čtecích zařízení je jejich mnohostranné použití.

Dále lze čtecí zařízení rozdělit podle toho, zda čtou sériově (čtecí pero) nebo paralelně (scannery, ruční pistole). Při paralelním čtení je celý kód přečten najednou. Čtecí zařízení využívá při čtení kódu světlo jako nosič informace. Používá se světlo ve viditelném (640 - 660 nm) i infračerveném (940 nm) pásmu. Infračervené (neviditelné) pásmo je používáno zejména pro utajení přenášených informací.

Dekodéry se dělí na externí, interní a integrované. Externí dekodéry jsou samostatná zařízení a jsou nejevšestrannější a nejkvalitnější. Mají ze všech dekodérů nejvíce funkcí, rozpoznávají nejvíce typů kódů a disponují většinou několika výstupními rozhraními. Dá se k nim obvykle připojit více čtecích zařízení (i více druhů) v jeden okamžik. Interní dekodéry jsou umístěny přímo na základní desce celého zařízení. Mají menší možnosti než externí dekodéry co se všech parametrů týče. Dekodéry integrované jsou se čtecím zařízením v jednom „obalu.“ Mají jeden typ výstupu (rozhraní), ale jsou specifické uživatelským komfortem (vysoká ergonomie) a jsou proto nejvhodnější do mobilních zařízení.

3.2 Typy čtecích zařízení a podporovaná rozhraní

Čtecí zařízení je možno rozdělit do následujících kategorií:

- CCD snímače
- laserové snímače
- čtecí pera
- šterbinové snímače (snímače karet)
- průmyslové snímače

- přenosné (paměťové) snímače
- stacionární snímače

Rozhraní podporovaná snímači (popř. dekodéry) jsou povětšinou sériová rozhraní RS232, RS422, RS485, TTL, klávesnicové rozhraní (PS/2) nebo USB. Zařízení v nejnižších cenových relacích podporují zejména rozhraní klávesnicové, RS232 a USB. Klávesnicové rozhraní má nespornou výhodu v tom, že není zapotřebí žádných úprav v softwaru. Speciální a zákaznické snímače podporují mnoho dalších různých rozhraní (OCIA, IBM 468X/469X ad.). Nově podporovaným a perspektivním rozhraním je bezdrátové připojení pomocí Wi-Fi.

CCD snímače čtou čárový kód přiblížením štěrbinou se světelnými diodami. Nevyžaduje se přímý kontakt s povrchem. Tyto snímače jsou přesné a rychlé. Většina snímačů se dá připojit ke klávesnici s možností zadávání údajů jak snímačem, tak klávesnicí. Čtení kódu je možné z bezprostřední blízkosti (max. vzdálenost může být cca 50 cm). U CCD snímačů je omezena maximální šířka čárového kódu od 60 mm do cca 120 mm.

Laserové snímače jsou využívány převážně v POS aplikacích (Point Of Sale). Jsou na ně kladeny vysoké nároky na produktivitu. Možnosti připojení k dalším zařízením mají stejné jako CCD snímače. Jsou schopny číst kódy až ze vzdálenosti deseti metrů.

Štěrbinové snímače slouží ke čtení karet s umístěným čárovým kódem. Snímání pobíhá většinou v infračerveném světelném pásmu.

Průmyslové snímače jsou využívány například v pásové výrobě v podobě velice výkonných holografických snímačů a tunelů.

Přenosné (mobilní, paměťové) snímače umožňují přenos dat bez okamžitého připojení kabelem k počítači. Svou velikostí připomínají větší kalkulačku, mobilní telefon nebo kapesní počítač a jsou vybaveny numerickou nebo alfanumerickou klávesnicí. Komunikace s počítačem probíhá buď po připojení kabelem nebo infračerveným rozhraním. U tohoto typu snímačů musí být dobře vyřešen způsob napájení a bezpečnost uložených dat. Zvláštním typem přenosných zařízení jsou ruční bezdrátové CCD nebo laserové snímače, kde komunikace probíhá přes radiofrekvenční rozhraní nebo je komunikace zahájena po založení čtecího zařízení do dekodéru ve formě stojánku. Pokročilejšími mobilními snímači jsou tzv. datové mobilní terminály, které se od ostatních vzhledem příliš neliší, ale disponují vlastním operačním systémem (např. DOS, Windows CE, Palm OS, ...). Snímač čárového kódu (nebo i jiný, např. magnetických karet) v je v mnoha případech u těchto přístrojů jedním z doplňků a ne integrovanou součástí.

Stacionární snímače (viz. také průmyslové snímače) jsou aplikovány ve formě čtecího zařízení zabudovaného např. v prodejním pultu a samostatného dekodéru. Jedná se o nejvýkonnější běžné snímače s možností vícesměrového čtení kódů. Jejich velkou výhodou je, že jejich obsluha má volné obě ruce pro manipulaci se zbožím.

Některé typy snímačů umožňují i dodatečné uživatelské programování mnoha funkcí (typ kódu, délka, meziznaková prodleva, volba jazyka, rychlost přenosu, nahrazení znaků atd.) nebo dokonce nových samostatných aplikací (datové mobilní terminály s vlastním operačním systémem). Některé typy umožňují i čtení čárových kódů přímo z monitoru počítače.

3.3 Referenční čtečka

Pro účely testování aplikace a pro její budoucí použití byla zakoupena ruční čtečka čárových kódů EZ One Shot, typ MT9060/4. Připojuje se přes rozhraní USB a v operačním systému je emulována jako USB klávesnice. Čtečka umožňuje snímat čárové kódy o maximální šířce 3,5 palce. Pro snímání se používá CCD scanner, který zvládá přesné načítání čárových kódů s velmi vysokou hustotou. Čtečka podporuje následující standardy: Code 39, UPC-A, UPC-E, Code 128, EAN-13, EAN-8, Interleaved 2 of 5, Codabar, Full ASCII Code 39, MSI/Plessey, Industrial 2 of 5, Code 93, BC412, Code 11, UPC/EAN. Dvourozměrné čárové kódy nejsou podporovány. Čtečku lze bez problémů používat pod všemi operačními systémy podporující připojení klávesnice přes USB.

Emulace čtečky jako klávesnice má svoje výhody, ale i nevýhody. Hlavní výhodou je, že nepotřebujeme ke zprovoznění čtečky žádný ovladač, postačí univerzální ovladač pro připojení USB klávesnice, kterým disponuje většina novějších operačních systémů. Na druhou stranu aplikace nedokáže rozlišit vstup z klávesnice od vstupu ze čtečky, což může být v některých případech potřeba. Aplikace nerozlišující vstup většinou počítá s interakcí uživatele, např. přesunutí kurzoru do textového pole. Každá taková interakce je spojena s určitou časovou ztrátou a při zadávání velkého množství čárových kódů najednou může hrát tato ztráta významnou roli.

Dalším sporným bodem je nastavení parametrů čtečky. Většina moderních čteček disponuje celou řadou parametrů např. maximální délka čárového kódu nebo nastavení podpory čtení pro konkrétní standard čárového kódu. V případě emulace čtečky klávesnicí neexistuje způsob jakým může aplikace s čtečkou komunikovat a nastavovat změny programově. Nastavení se pak provádí pomocí servisních čárových kódů, které jsou uvedeny v manuálu dodávaném ke čtečce. Před použitím konkrétní aplikace se musí čtečka podle tohoto manuálu nastavit.

4 Tisk čárových kódů

Pro rychlé a spolehlivé přečtení čárového kódu je nutné zajistit jeho kvalitní vtištění. Jednou z podmínek je, že kód musí být v definovaném tolerančním pásmu, jinak bude čtení nejisté. Mezi kritéria výběru tiskové metody patří na jaké médium bude čárový kód tisknut, v jaké hustotě a počtu, jaká by měla být životnost kódu, kde a jak se bude s kódem pracovat a jaká by měla být maximální cena.

Pro tisk na tiskárnách řízených počítačem můžeme použít mnoho druhů tiskáren. Následuje jejich přehled a krátký popis vhodnosti pro tisk čárových kódů.

- bubnové tiskárny – vysoká kvalita tisku, vhodné i pro vysokou hustotu
- jehličkové tiskárny – nespolehlivé pro tisk čárových kódů
- laserové tiskárny – výkonné s vysokou kvalitou tisku i pro vysokou hustotu
- termotiskárny – nevhodné pro vysokou hustotu, tiskne se termocitlivý papír, který je citlivý na světlo a teplo a výsledek je tak nestabilní, velmi dobrá kvalita tisku, tyto tiskárny bývají specializovány na tisk čárových kódů
- termotransfer tiskárny – barevný tisk, vysoká kvalita, vyšší náklady na tisk

5 Návrh řešení

Cílem práce je vytvořit agendu hodnocení studentů založenou na čárových kódech. Podnětem pro tvorbu této agendy bylo zautomatizování procesu zadávání známek studentů při písemných zkouškách a zvýšení anonymity studenta v rámci zkoušky. Aby mohl celý systém efektivně fungovat, je nutná spolupráce s již existující univerzitní studijní agendou (dále STAG). Základním elementem systému je čárový kód, který musí jednoznačně identifikovat studenta a termín konkrétní zkoušky, a to v rámci celé univerzity. Z důvodu jednotné celouniverzitní identifikaci se vyloučí problémy při ztrátě testu a dohledávání jeho vlastníka. Systém bude rovněž úzce spjat i se STAGem.

Vlastní řešení problému tedy vychází z následujících bodů:

- provázání aplikace se STAGem
- volba řetězce pro zakódování, otázka zabezpečení
- způsob provázání čárového kódu s testem
- způsob evidence hodnocení
- výstupy a propagace výsledků hodnocení

5.1 Navázání aplikace na systém STAG

Systém STAG je informační systém pro evidenci studijní agendy vysoké školy nebo univerzity. V současné době je postaven na databázovém systému Oracle verze 10g. Pro naši aplikaci potřebujeme získat přístup k tabulkám studentů a zkouškových termínů. Konkrétně pro vytvoření čárových kódů potřebujeme seznam studentů zapsaných na určitý termín, seznam termínů pro určitý předmět a seznam všech studentů zapsaných na určitý předmět. Po konzultaci se správcem STAGu byl vytvořen pohled (VSTUDENTI.ZNAMKY), který všechny tyto potřebné informace poskytuje. K databázi se připojíme v roli učitele a pohled se naplní daty z příslušných tabulek STAGu. Samotná struktura databáze STAGu nás tedy nemusí zajímat, neboť záznamy z pohledu budeme pouze importovat do naší aplikace.

Struktura pohledu VSTUDENTI.ZNAMKY(nejdůležitější položky)

Jmeno jméno studenta

Prijmeni příjmení studenta

Os_cislo osobní číslo studenta

Rok_platnosti školní rok pro něž byla vypsána zkouška

Prac_zkr zkratka katedry

Zkr_predm zkratka předmětu

Semestr semestr pro něž byla vypsána zkouška

Znamka konečná známka zapsaná sekretariátem katedry na STAG

Termidno celočíselný identifikátor testu - maximálně 10 cifer

Zkr_budovy zkratka budovy, ve které probíhá či proběhla zkouška

Cislo_mist číslo místnosti v rámci budovy

Typ_terminu určuje, zda se jedná o zkoušku nebo zápočet

V pohledu jsou pouze studenti, kteří jsou zapsáni na zkoušku předmětu, pro kterou je přihlášený učitel vyučujícím nebo cvičícím. Pohled tedy lze využít pro následné importování :

- kateder a předmětů, které učitel vyučuje a existují pro ně zkuškové termíny, na kterých je přihlášen alespoň jeden student
- studentů, kteří mají zapsán předmět a zúčastnili se alespoň jednoho zkuškového termínu
- studentů přihlášených na zkoušku předmětu definovaném katedrou, předmětem a školním rokem

Tyto záznamy postačují k vytvoření čárových kódů pro studenty zapsané na určitém termínu zkoušky. Navíc je přístup omezen pouze pro učitele, který daný předmět vyučuje nebo cvičí.

5.2 Volba řetězce pro zakódování

Při volbě řetězce musíme vycházet ze záznamů, které nám poskytuje databáze STAGu. Jednoznačným identifikátorem studenta v rámci univerzity je osobní číslo. Jedná se o identifikátor skládající se z písmene specifikujícího fakultu a maximálně šesti cifer upřesňujících studenta v rámci fakulty. Pokud místo úvodního znaku použijeme jeho index v ASCII tabulce, získáme identifikátor o délce osmi cifer. Zbývá nám tedy zanechat do kódu informaci jednoznačně identifikující termín. V systému STAG se každému termínu přiděluje desetimístný unikátní identifikátor a pokud jej spojíme s osobním číslem, dostaneme celočíselný identifikátor o celkové délce 18 číslic.

Nyní potřebujeme zvolit vhodný standard čárového kódu. Tento výběr je omezen především délkou kódovaného identifikátoru. Žádný ze standardů s pevnou délkou však nepodporuje přesně 18 cifer. Možným řešením by bylo použít dva kódy s pevnou délkou, což by ovšem znamenalo složitější proces při čtení čárového kódu. Použijeme proto standardu s proměnou délkou. Zvolíme nejpoužívanější a nejuniverzálnější standard CODE39.

Kódovací tabulka standardu CODE39 se skládá ze 43 znaků. Abychom plně využili kódovací tabulku, převedeme 18 cifer identifikátoru z desítkové soustavy do soustavy o základu 42. Do kódování nebudeme uvažovat mezeru z důvodu lepší čitelnosti výstupního řetězce. Následným převodem zmenšíme šířku výsledného čárového kódu, což nám umožní snížit hustotu zápisu a tím zvýšit spolehlivost čtení. Délku identifikátoru o 18

cifrách v soustavě se základem 42 určíme jednoduchým výpočtem:

$$\begin{aligned} 42^k &= 10^{18} \\ k &= \frac{18}{\log(42)} \\ k &= 11,0889 \end{aligned}$$

Převedený identifikátor má délku 12 znaků.

Uvedený způsob je jedním z možných řešení. Pokud bychom chtěli zvýšit bezpečnost kódu a zmenšit délku kódovaného řetězce, bylo by vhodnější použít binární soustavy. Písmeno specifikující katedru je vyjádřeno dvěma číslicemi desítkové soustavy. Za předpokladu, že by byla využita všechna velká písmena abecedy, stačilo by nám k zakódování pouze 5 bitů. Tím bychom ušetřili jeden bit a délka celého identifikátoru by klesla na 11 znaků. Přidáním dalšího znaku, který by byl pro každého student generován, bychom navíc zvýšili bezpečnost.

5.3 Zabezpečení čárového kódu proti zneužití

Čárový kód by měl být zabezpečen proti zneužití. Student by totiž mohl nahradit čárový kód vlastním a zaměnit tak svoji identitu. Pokud by student neznal skladbu identifikátoru, byla by analýza značně komplikovaná. Převod složek identifikátoru do soustavy o základu 42 tak ztíží práci případnému „útočníkovi“. Pokud by se mu přece jen podařilo nalézt základ soustavy a dekodovat strukturu identifikátoru, potřeboval by ještě identifikátor termínu, na kterém chce změnu identity provést. Ten lze zjistit jen analýzou URL řetězce při prohlížení obsazení zkoušek přes webové rozhraní STAGu. Celý systém tedy není zcela zabezpečený, ale převod do soustavy o základu 42 je dostačujícím bezpečnostním prvkem.

Zabezpečení by se dalo vyřešit i jiným způsobem. Mohli bychom využít změny mapování kódovací tabulky standardu CODE39 na jednotlivé číslice soustavy o základu 42. Pozměněná mapovací tabulka by se pak musela uložit, aby mohlo dojít ke zpětnému dekódování. Vytváření mapovací tabulky by zajišťoval generátor náhodných čísel, takže v důsledku by případný útočník musel projít 42! kombinací. Uvedené řešení ale není příliš vhodné, protože při ztrátě mapovací tabulky by se nemohl dohledat test ani jeho vlastník.

5.4 Provázání kódu s testem a tisk čárových kódů

V zásadě existují dva přístupy jak aplikovat čárový kód na test. Čárový kód může být přímo součástí sestavy pro tisk zadání testu. Takové řešení by ovšem vyžadovalo centrální úložiště pro všechny testy a jeho provázání se systémem hodnocení. Takové úložiště však není k dispozici, a proto použijeme druhé řešení – tisk čárových kódů na etikety. Student tedy použije etiketu místo vlastního podpisu.

Pro zjednodušení procesu rozdělování čárových kódů mezi studenty je vhodné tisknout čárové kódy na řezané štítky. Řezané štítky se distribují ve formátu A4. Obecně lze arch s těmito štítky popsat následujícími parametry: počet řádek štítků, počet

sloupců štítků a okraj stránky. Tisk je velice náročný na přesnost, posunutí stránky uvnitř zásobníku tiskárny znamená většinou znehodnocení celé strany sestavy čárových kódů. Tím se také na jedné etiketě může objevit více čárových kódů, což znemožní správné načtení kódu.

5.5 Způsob evidence hodnocení

Univerzitní agenda STAG neumožňuje zadávat jiné hodnocení předmětu než známku studenta. Systém pro evidenci hodnocení musí být více komplexní, tak aby umožňoval do hodnocení zanechat informaci i o testu. Evidence hodnocení čárovými kódy pracuje pouze s písemnými testy, musí se tedy nějak zobecnit hodnocení testu. Nejjednodušším způsobem je rozdělit test na otázky a stanovit maximální počet bodů. Tento počet bodů se pak rozdělí mezi otázky dle jejich obtížnosti. Dále se určí 4 různé intervaly pro 4 stupně známek - výborně, velmi dobře, dobře a neprospěl. Při zadávání hodnocení se ohodnotí každá otázka zvlášť a podle výsledné sumy bodů a intervalu se určí konečná známka. Aby systém fungoval, musí mít každý zkuškový termín přidělenou šablonu testu s nadefinovanými otázkami, počtem bodů k jednotlivým otázkám a intervaly hodnocení. K hodnocení studenta pro určitou zkoušku je rovněž vhodné přidat poznámku, která může být užitečná např. při reklamaci výsledku.

5.6 Výstup a propagace výsledků hodnocení na univerzitním portálu

Hodnocení se musí nejen evidovat, ale také je potřeba výsledky určitým způsobem zveřejnit. A to navíc ve formě, která umožňuje výsledky dále zpracovávat (vytvářet různé statistiky apod.). Nejvhodnějším místem pro propagaci výsledků je univerzitní portál. Ten sice spolupracuje se STAGem, avšak to pro naši evidenci hodnocení nepřináší žádné výhody. Zápis známek na STAG zajišťuje sekretariát příslušné katedry většinou až před koncem semestru. Většina studentů ale chce znát výsledky zkoušek v co nejkratší době. Součástí portálu je courseware, tedy místo, kam lze ukládat různé soubory, studijní materiály a další informace týkající se jednotlivých předmětů. A právě sem je vhodné ukládat výsledky zkoušek. Popis systému courseware spolu s návodem pro přidání výsledků testů je volně přístupný po přihlášení na univerzitní portál v sekci předměty.

6 Realizace aplikace

Aplikace byla vyvíjena pod .NET framework 2.0, v programovacím jazyce C# a prostředí Visual Studio 2005. Cílová aplikace je určena pro systém Windows a při její tvorbě aplikace bylo zapotřebí vyřešit následující problémy:

- připojení k databázi systému univerzitní agentury STAG
- tisk čárových kódů
- správa lokálního uložení

6.1 Připojení k databázi Oracle

Související třídy :

OracleAccessor třída pro přístup k databázi Oracle, která obsahuje metody pro získání dat z databáze a konverzi do lokálních struktur

Databázový systém Oracle je produktem firmy Oracle corporation. V dnešní době se jedná o jeden z nejpoužívanějších databázových systémů. Databáze Oracle používá relační model uložení dat. Relace mezi datovými objekty vznikají na základě požadavků uživatele. Pro komunikaci se využívá jazyka SQL. Programový kód (příkazy SQL, skripty nebo celé programy) je v systému uložen nezávisle na struktuře dat.

Pro připojení k databázovému serveru Oracle je zapotřebí klientského programu. Pro vývoj a nenáročnou aplikaci poskytuje Oracle bezplatnou verzi klienta – Oracle Database XE klient. Po instalaci zabírá klient na disku přibližně 80MB a obsahuje všechny základní balíky nutné pro připojení k Oracle databázi. Tento klient je vhodný i pro naši aplikaci, stačí nám pouze přístup k jednomu pohledu do databáze a nepotřebujeme využívat žádných pokročilejších funkcí.

V prostředí .NET se pro připojení k Oracle databázi používá ADO.NET¹. ADO.NET poskytuje pro připojení k databázím různých druhů vlastní providery – třídy, ve kterých je implementováno spojení s databází.

6.2 Tisk čárových kódů v aplikaci

Třídy, které zajišťují tisk čárových kódů a nastavení jednotlivých parametrů tisku:

BarcodePrinter.cs - třída zajišťující tisk sestav čárových kódů dle nakonfigurovaných parametrů

BarcodePrintDialog.cs - třída pro konfiguraci parametrů tiskových sestav čárových kódů tisknutých na řezané štítky

Tisk čárových kódů v aplikaci je řešen pomocí funkcí GDI+. Základem je font čárového kódu.

¹soubor tříd pro přístup k datům v technologii .NET

6.2.1 GDI+

Jedná se o kolekci tříd, které poskytují přístup ke grafickému rozhraní systému Windows. Umožňuje uživateli zobrazit informace na libovolném zobrazovacím zařízení připojeném k systému (monitor, tiskárna, . . .), aniž by musel sám znát detaily o výstupním zařízení. Volání metod tříd je převedeno na volání příslušných ovladačů. GDI+ odděluje aplikaci od grafického hardwaru a umožňuje tak vytvářet aplikace nezávislé na zařízení.

Pro definování cíle výstupu se používá speciální objekt zvaný kontext zařízení. Ten ukládá informace o ovladači zařízení a je schopen převést volání metod grafického rozhraní do instrukcí pro ovladač. Rovněž umožňuje získat informace o zařízení, jako např. v případě tiskárny informace o rozlišení, pevných okrajích tisku apod. V .NET framework je kontext zařízení zabalen do třídy **Graphics**.

6.2.2 Font čárového kódu

Font je předpis, který specifikuje vzhled jednotlivých tištěných znaků. Rodina fontů obsahuje několik řezů písma, které jsou si navzájem podobné. První řez je vždy základní a ostatní jsou vyznačovací (kurzíva, tučné písmo, tučná kurzíva). Velikost písma se určuje prostřednictvím výšky písma. Výška písma je měřena v bodech, bod reprezentuje 1/72 palce. Celková výška se skládá z dílčích výšek, např. výška účají, výška akcentu. To nás ale nemusí zajímat, stačí pouze definovat hlavní výšku a rodinu fontu, o zbytek se GDI+ postará samo.

Rodina fontů TrueType je definována pomocí ploch a křivek, které precizně popisují jednotlivé znaky v matematickém smyslu. Díky tomu lze tisknout znaky v libovolné velikosti. Font čárového kódu definuje pro každý znak odpovídající skupinu mezer a čar. Aby se dosáhlo vysoké přesnosti při tisku různých velikostí čárových kódů, používají se TrueType fonty. Font čárového kódu lze vytvořit pro libovolný čárový kód. Pro převod na čárový kód stačí mít tedy pouze řetězec, který chceme na čárový kód převést a tento řetězec vytisknout odpovídajícím fontem čárového kódu.

Řetězec musí dodržet různé zásady specifické pro každý typ čárového kódu. Musí se v něm vyskytovat pouze znaky podporované čárovým kódem a musí být obklopen start a stop znaky, aby při zpětném snímání čtecího zařízení tento kód rozpoznalo. Výhodou fontů čárového kódu je, že se dají využívat v libovolné aplikaci, která podporuje TrueType fonty. Font stačí v operačním systému nainstalovat a od té doby je připraven k použití.

V naší aplikaci budeme tisknout čárové kódy standardu CODE39. Použijeme volně šiřitelný font od společnosti IDAutomation – IDAutomationHC39M. Podrobné informace o tomto fontu lze nalézt v [5]. Nejdůležitějším parametrem u fontů čárových kódů je závislost šířky světelného modulu na výšce fontu čárového kódu. Podle toho můžeme vidět, jakou má čárový kód hustotu pro konkrétní výšku písma.

Specifikace pro kód CODE39 jsou jednoduché, stačí dodržet kódovací tabulku [viz. kapitola 1.4] a přidat znak "*" na začátek a konec řetězce jako start a stop znak.

6.2.3 Postup při vytvoření řetězce pro tisk fontem čárového kódu

Nejprve upravíme osobní číslo studenta tím, že odstraníme písmeno ze začátku a nahradíme jej odpovídajícím indexem do ASCII tabulky. V případě, že má studijní číslo méně než 8 cifer, přidáme nulu zleva. Spojíme identifikátor s osobním číslem a výsledek převedeme tak, abychom dostali řetězec specifikující identifikátor v soustavě o

Výška písma	Světelný modul X [mm]
48	1.57
36	1.18
24	0.78
20	0.65
16	0.52
12	0.39
8	0.26
6	0.19

Tabulka 3: Charakteristika fontu IDAutomationHC39M.ttf

základu 42. Jako cifry soustavy použijeme jednotlivé znaky kódovací tabulky kromě znaku mezery, to kvůli lepší čitelnosti výsledného řetězce. Převod probíhá postupným dělením spojeného identifikátoru číslem 42 a zjišťováním zbytku. Potřebujeme celkem 12 znaků z důvodu, že maximální možné číslo vyjádřitelné spojením identifikátoru termínu a převedeného osobního čísla je 10^{18} . To vyjádříme pomocí soustavy o základu 42 prostřednictvím 12 cifer. Následuje příklad použití uvedeného postupu.

osobní číslo = A05422
identifikátor termínu = 166491

upravené osobní číslo = 06505422
spojený identifikátor = 16649106505422

cílový řetězec po převodu = 60EQS69U1000
řetězec pro tisk č.k = *60EQS69U1000*

6.2.4 Řezané štítky

Tisk na řezané štítky vyžaduje velkou přesnost při rozmísťování čárových kódů do tiskové sestavy. Jsme omezeni výškou a šířkou jednoho štítku. Obecně platí, že čárové kódy s menší hustotou jsou více spolehlivé pro načtení. Otázkou je, jakým způsobem vytisknout čárový kód tak, abychom využili celou plochu štítku a tím dosáhli co nejmenší hustoty čárového kódu. Předpokládáme, že rozměry štítků si uživatel může nastavit, proto musíme všechny parametry pro tisk stanovit dynamicky za běhu programu.

Rozhraní GDI+ zavádí funkci, která vrací šířku a výšku oblasti, jenž zabírá řetězec tisknutý určitým fontem o dané velikosti. Pomocí této funkce realizujeme jednoduchý algoritmus pro stanovení maximální výšky čárového kódu v závislosti na šířce etikety. Hledáme tedy takovou výšku fontu čárového kódu, při které by se čárový kód svou šířkou vešel do přesně vymezené oblasti štítku. Realizace tohoto algoritmu je ve třídě **GraphicMisc.cs**, metoda *GetMaximumTextHeight()*.

V některých případech může být šířka etikety podstatně menší než výška čárového kódu. Výsledný čárový kód je pak úzkým proužkem s maximální hustotou. To může znesnadňovat čtení např. při lokálním poškození kódu. V tom případě se nabízí dvě řešení.

Použít jiný typ fontu, který má vysokou hustotu i pro větší výšky písma, nebo vertikálně poskládat jednotlivé čárové kódy nad sebe, čímž výšku zvětšíme. V aplikaci používám druhý způsob, řešení je ve třídě **GraphicMisc.cs**, metoda *FillRectangleWithBarcode()*.

6.3 Lokální uložení

Při návrhu systému bylo potřeba učinit rozhodnutí, jakým způsobem ukládat data importovaná ze systému STAG a vlastní data pro hodnocení studentů. Možným řešením by bylo vytvořit databázový server a data ukládat na něj. To by mělo výhodu v tom, že data by byla uložena centralizovaně a mohl by k nim přistupovat kdokoli. Pro toto řešení bychom potřebovali vlastní server, který by musel někdo spravovat. Pro ukládání hodnocení nepotřebujeme velké množství tabulek, proto je lepším řešením ukládat data do lokálního uložení a omezit tak přístup pouze pro uživatele aplikace na konkrétním stroji.

Lokální uložení je reprezentováno souborem uloženým na pevném disku spolu s aplikací. Do tohoto souboru jsou používána data ukládána ve formátu XML. Na formátu v jakém budou data uložena nezáleží, samotný běh aplikace tím nebude ovlivněn. Pokud však použijeme XML, získáme výhodu platformové nezávislosti při případném dalším zpracování datového souboru.

Všechna data jsou při spuštění aplikace načtena do struktur v paměti. Opětovné přepsání souboru aktualizovanými daty proběhne až před samým ukončením programu. Proces načítání a ukládání je prováděn pomocí tzv. serializace. Ta slouží pro uložení obrazu instance objektu do datového proudu (stream). Cílem datového proudu může být např. soubor nebo blok operační paměti. Způsoby uložení instance objektu jsou různé, pro naši aplikaci jsem použil serializaci do XML. Třídy pro XML serializaci jsou dostupné ve jmenném prostoru *System.Xml.Serialization*.

Nejdůležitější třídou v tomto jmenném prostoru je *XmlSerializer*. Pro serializaci objektu potřebujeme vytvořit instanci objektu *XmlSerializer* a specifikovat výchozí objekt. Dále je nezbytné vytvořit instanci objektu datového proudu pro zápis do souboru. Posledním krokem je zavolání metody *Serialize()* objektu *XmlSerializer*. Pomocí parametrů této metody předáme serializovaný objekt a stream. Do streamu se implicitně ukládají všechny public položky serializovaného objektu.

Opačným postupem k serializaci je deserializace, kdy se na základě obsahu streamu vytvoří instance tříd v paměti a naplní se její public položky.

Veškeré objekty určené pro serializaci jsou umístěny ve třídě *LocalRepositoryAccessor* jako public položky. *XmlSerializer* se volá přímo nad touto třídou v metodách třídy *LocalRepositoryAccessor* *Load()* a *Save()*. Výsledkem serializace je tedy pouze jeden soubor, který obsahuje všechna data potřebná pro běh aplikace.

Zásadní nevýhodou serializace je to, že se všechna data ukládají a načítají do paměti najednou. Prakticky lze ovlivnit pouze zdroj dat pro ukládání a cílový soubor, vše ostatní je ponecháno na .NET frameworku. Pokud by data ukládaná do uložení byla příliš rozsáhlá, mohlo by dojít k zahlcení operační paměti počítače a zpomalení běhu celého systému. Na druhou stranu serializace podstatně usnadňuje práci vývojáři.

7 Závěr

Výsledkem práce je funkční aplikace, která umožňuje konfigurovatelný tisk sestav čárových kódů na řezané samolepicí štítky a zpětně zadávání hodnocení studentů pomocí takto vytvořených čárových kódů. Aplikace spolupracuje s vývojovou databází na Oracle serveru studijní agendy STAG.

Aplikace byla vyvíjena v prostředí .NET framework 2.0 s využitím možností moderního jazyka C# a je určena pro operační systém Windows. Umožňuje exportovat výsledky hodnocení do HTML tabulky, což plně postačuje pro zveřejnění výsledků na internetu i v rámci univerzitního portálu. Pro vytvoření složitějších statistik z hodnocení lze bez potíží importovat HTML tabulku do některého z tabulkových procesorů. Data s hodnocením studentů jsou ukládána do souboru na lokálním disku uživatele této aplikace. Výsledky hodnocení tedy nejsou ukládány na vlastní centrální databázový server a aplikace rovněž neumožňuje export známek do systému studijní agendy STAG. Pro tento účel je určena funkce generování tiskových sestav se známkami studentů zapsaných na určitý předmět. Tyto sestavy se pak mohou využít pro zadávání známek např. sekretariátem katedry.

Evidence hodnocení studentů čárovými kódy představuje velmi komplexní problém. Vytvořená aplikace sice dokáže importovat data ze STAGu, ale chybí zpětný zápis známek do systému STAG. Evidence by měla být centrálně zpřístupněna pro všechny uživatele, a tak by jedním z možných řešení do budoucna bylo použití univerzitního portálu.

8 Seznam zkratek

STAG	informační systém určený pro evidenci studijní agendy vysoké školy nebo univerzity
Start a stop znak	znaky které určují začátek a konec čárového kódu, používají se pro detekci druhu čárového kódu čtecím zařízením
CODE39	standard čárového kódu, vyniká jednoduchostí a spolehlivostí při čtení
.NET framework	platforma pro vývoj aplikací pro operační systém Windows
HTML	HyperText Markup Language – jazyk pro vytváření stránek v systému world wide web
ASCII	American Standard Code for Information Interchange – tabulku kódů, která definuje znaky anglické abecedy a jiné znaky používané v informatice
XML	eXtensible Markup Language – obecný značkovací jazyk

Literatura

- [1] Adriana Benadiková, Štefan Mada, Stanislav Winlich: *Čárové kódy – automatická identifikace*, Grada, 1994.
- [2] Bc. Tomáš Legát, *Integrace čárového kódu do prostředí Oracle Developer R6*, diplomová práce, 2003.
- [3] Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner, Allen Jones : *Professional C# 2005*, Wrox Press, 2006.
- [4] Brian Noyes : *Data Binding with Windows Forms 2.0: Programming Smart Client Data Applications with .NET*, Addison Wesley, 2006.
- [5] IDAutomation, barcode font : <http://www.idautomation.com/fonts/free/>

A Příloha – Uživatelská příručka

A.1 Popis uživatelského rozhraní programu

Po spuštění aplikace se objeví hlavní formulář se třemi záložkami a jedním tlačítkem v horním menu. Jednotlivé záložky formuláře slouží ke zobrazení podformulářů s následujícími funkcemi:

- Správa termínu - zajišťuje import předmětů, termínů zkoušek a studentů z databázového serveru univerzitní agendy
- Vstup hodnocení - slouží pro vstup hodnocení pomocí čtečky čárových kódů
- Export výsledků - formulář pro export výsledků hodnocení jednotlivých zkoušek

A.2 Správa termínů

V levé části formuláře je umístěn strom pro výběr katedry a předmětu. Pod tímto stromem lze pomocí rozbalovacího seznamu a textového pole zvolit příslušný školní rok. Veškeré další operace budou závislé právě na tomto školním roce. Posledními funkčními položkami v levé části formuláře jsou dvě přepínací tlačítka sloužící jako filtr pro strom předmětů - zobrazit vše a pouze importované. Ta určují, zda aplikace ve stromu předmětů zobrazí všechny předměty, které učitel vyučuje a dosud je neimportoval do lokálního uložení, nebo zobrazí pouze předměty, které již byly naimportovány. Pravá část formuláře obsahuje dvě tabulky. Horní tabulka zobrazuje všechny termíny vypsané pro předmět. Červeně jsou v ní označeny termíny, které už proběhly, zeleně pak termíny, které budou teprve následovat. Jednotlivé položky tabulky jsou řazeny vzestupně podle data termínu. Při volbě položky v horní tabulce se spodní tabulka naplní seznamem studentů zapsaných na příslušné zkoušce.

Ve spodní části formuláře jsou zleva umístěna 4 tlačítka:

- Importování předmětu – po stisknutí je předmět i se studenty zapsanými na tento předmět přesunut do souboru lokálního uložení. Po této akci je možné dále manipulovat s jednotlivými termíny zkoušek.
- Aktualizovat data – provede opětovné načtení dat z databáze studijní agendy. Tím lze tedy využít k aktualizaci seznamu studentů na zkuškovém termínu.
- Přiřadit test – zobrazí dialog pro přiřazení testové šablony konkrétnímu zkuškovému termínu, jenž je zvolen v tabulce termínů zkoušek. Změnu přiřazeného testu lze provádět pouze tehdy, nebyl-li dosud hodnocen žádný student zapsaný na zkuškovém termínu.
- Tisk sestavy čárových kódů – zobrazí dialog s nastavením tisku sestavy čárových kódů pro termín zkoušky, který je zvolen v tabulce termínů zkoušek.

A.2.1 Přiřazení testu

Po stisku tlačítka přiřadit test se zobrazí dialog s tabulkou existujících testů. Uživatel zvolí test, který chce zkoušce přiřadit a stiskne tlačítko "OK". Ve spodní části dialogu je panel navigátoru. Na něm jsou umístěna tři tlačítka. První tlačítko zleva umožňuje vytvoření nového testu. Další dvě tlačítka zabezpečují operace mazání a aktualizace již vytvořeného testu.

Po stisku tlačítka pro vytvoření testu se zobrazí dialog, v němž definujeme název testu, počet otázek testu a celkový počet bodů testu. Dále jsou zde 3 textová pole pro vstup intervalů hodnocení určující 3 známky. Horní meze intervalů se dopočítávají automaticky a zobrazují se vedle textových polí pro vstup. Pod definicí intervalu je tlačítko s popisem výchozí, které umožňuje vytvořit intervaly automaticky ze zadaného celkového počtu bodů. Posledním tlačítkem v dialogu je bodování otázek.

Stiskem tlačítka bodování otázek může uživatel dodefinovat konkrétní rozdělení bodů mezi jednotlivé otázky a přiřadit otázkám popisky. Děje se tak ve vlastním dialogu, kde je zobrazena tabulka s již existujícími otázkami. Pod ní jsou umístěna dvě textová pole pro zadání popisku a hodnocení otázky, dále pak 5 tlačítek zabezpečujících akce vkládání, mazání, aktualizace otázek a vytvoření výchozích otázek. Všechny změny se pak vizuálně propagují do tabulky s otázkami. Pokud se počet otázek nebo počet bodů po změně bodování neshoduje s počtem v dialogu pro nastavení testu, aplikace uživatele na tuto skutečnost upozorní.

A.2.2 Tisk čárových kódů

Dialog pro nastavení tisku čárových kódů je přístupný přes tlačítko ve správě termínů. V levé části dialogu jsou zobrazeny položky pro nastavení jednotlivých parametrů čárových kódů, při tisku na formát A4 s řezanými samolepícími štítky.

- počet řádků – určuje celkový počet řádků řezaných štítků
- počet sloupců – určuje celkový počet sloupců řezaných štítků
- počet využitých řádků – určuje počet řádků využitých při tisku, řádky se zarovnávají na střed, pokud zadáme hodnotu 12 a celkový počet řádků bude 14, nebude poté využit první a poslední řádek
- vertikální korekce – posune celou tiskovou sestavu na stránce o zadaný počet milimetrů ve vertikálním směru, funguje v kladném i záporném smyslu
- horizontální korekce – posune celou tiskovou sestavu na stránce o zadaný počet milimetrů v horizontálním směru, funguje v kladném i záporném smyslu
- velikost okraje – umožňuje specifikovat velikost okraje stránky, tedy zmenšení aktivní tiskové oblasti v počtu milimetrů
- záhlaví – definuje text, který se objeví v horní části štítku nad čárovým kódem, možnosti jsou:
 1. jméno studenta
 2. osobní číslo studenta

3. řetězec čárového kódu

- pata – stejně jako v případě záhlaví, ale pro dolní část štítku
- styl tisku – určuje, jakým způsobem budou označeny čárové kódy v tiskové sestavě a existují dvě možnosti:
 1. pouze čárové kódy – na štítky vytiskne jen čárové kódy
 2. čárové kódy s popisem – před každou etiketu s čárovým kódem umístí iniciály studenta

Konfiguraci parametrů lze ukládat a k tomuto účelu slouží dvě tlačítka v levé dolní části dialogu. Po stisku tlačítka „Nová“ se zobrazí dialog s textovým polem pro zadání názvu konfigurace. Tlačítko „Smazat“ odstraní aktuální konfiguraci a způsobí přepnutí na některou z existujících konfigurací. Výběr konfigurace umožňuje rozbalovací seznam nad jednotlivými parametry konfigurace.

Změny konfigurace projevující se na vzhledu jednoho štítku lze kontrolovat v pravé horní části dialogu. Spolu s ukázkou vzhledu štítku se zobrazuje měřítko a rozměry štítku v milimetrech. V pravé dolní části můžeme vidět počet stran sestavy a zaškrtačací tlačítko pro tisk ohraničení jednotlivých štítků – označeno jako tisk mřížky.

Tlačítkem tisk zobrazíme standardní systémový dialog pro definování cílové tiskárny a intervalu tisknutých stránek. Místo samotného tisku můžeme také zvolit pouze zobrazení náhledu.

A.3 Vstup hodnocení

Uprostřed formuláře vidíme text s upozorněním, že se očekává vstup ze čtečky a také tlačítko pro manuální zadání čárového kódu z klávesnice. Použijeme-li pro vstup čtečku a zadaný čárový kód je platný, zobrazí se nový formulář na místo původního. Vlevo jsou zobrazeny základní informace o studentovi a zkoušce, vpravo pak vzhled štítku s čárovým kódem tak, jak by měl vypadat ve skutečnosti. V levé spodní části jsou zobrazeny jednotlivé otázky a k nim příslušná textová pole pro zadávání hodnocení. Vpravo je umístěno textové pole pro zapsání poznámky. Po stisku tlačítka „Uložit“ se hodnocení uloží do evidence. Uživatel také může provést po stisku příslušného tlačítka a zobrazení dialogu s textovým polem vstup čárového kódu manuálně z klávesnice. Zadá-li platný kód, zobrazí se stejný formulář jako v případě vstupu ze čtečky.

A.4 Export výsledků

Podobně jako u formuláře pro správu termínu je též na levé části exportu výsledků umístěn strom předmětů a komponenta pro zadání časového období. Přepínací tlačítka pro filtrování umožňují filtrovat studenty, kteří nemají zapsanu známku ve studijní agendě STAG. V pravé části formuláře se po zvolení katedry a předmětu zobrazí tabulka se seznamem studentů, kteří jsou zapsáni na příslušný předmět. U každého studenta je v tabulce uvedeno hodnocení a počet pokusů. Ve spodní tabulce se po výběru studenta zobrazí seznam všech zkoušek, které student v rámci předmětu absolvoval. Po dvojitém poklepání na jednu z těchto zkoušek se zobrazí dialog s hodnocením jednotlivých otázek a uloženou poznámkou. Ve spodní části dialogu jsou zprava umístěna dvě tlačítka. Jedno

z nich je pro tisk sestavy studentů zapsaných na předmět spolu s jejich hodnocením a druhé pro exportování výsledků zkoušek do HTML.

Tisk sestavy studentů zapsaných na předmět podléhá filtru. Jeho použitím se tak mohou vytisknout pouze studenti, kteří nemají zapsanou známku ve studijní agendě STAG. Před samotným tiskem se zobrazí dialog s náhledem sestavy a tisk se spustí stisknutím tlačítka se symbolem tiskárny v horní liště dialogu.

Po stisku tlačítka pro export do HTML se zobrazí dialog s tabulkou všech zkoušek, které proběhly v rámci zvoleného předmětu. Pod tabulkou zkoušek je seznam zaškrtačiacích tlačítek s možnostmi nastavení exportu. Tlačítkem export se zobrazí dialog pro volbu jména a umístění cílového souboru, do kterého bude uložena HTML tabulka. Potvrzením jména cílového souboru se export zahájí.

A.5 Připojení k databázi studijní agendy STAG

V případě, že se chce uživatel přihlásit na jiné uživatelské konto v systému STAG, nebo nebyl dosud k systému STAG přihlášen, může využít tlačítka v horním menu aplikace „STAG Login“. Po stisku tohoto tlačítka se zobrazí dialog pro připojení. V něm se musí vyplnit uživatelské jméno a heslo. Pokud se uživatel nepřipojuje k vývojové databázi STAGu, musí změnit i koordináty připojení k databázi. Není-li uživatel přihlášen, je vždy při pokusu aplikace o přístup k databázi STAGu vyzván k vyplnění přihlašovacích parametrů prostřednictvím dialogu pro připojení.

A.6 Instalace aplikace

Aplikace není distribuována s vlastním instalátorem, distribuuje se v archivu, který obsahuje tři soubory :

BP.exe hlavní spustitelný soubor aplikace

BP.exe.config konfigurační soubor programu

IDAutomationHC39M_Free.ttf font čárového kódu

Obsah archivu nakopírujeme do libovolné složky na disku. Soubor s fontem čárového kódu je nutné nainstalovat do systémového úložiště fontů. Instalaci provedeme přímo v systému Windows. V ovládacích panelech zvolíme Písma. Po zobrazení okna se seznamem písem, klikneme v horním panelu tohoto okna na položku „Soubor“ a následně „Instalace nového písma“. V dialogu vybereme soubor IDAutomationHC39M_Free.ttf a to ve složce, do které jsme nakopírovali obsah archivu a stiskneme tlačítko „OK“. Nyní by měl být font nainstalován v systému a můžeme ho využívat v libovolné aplikaci, která podporuje TrueType fonty. Program pro svůj běh předpokládá přítomnost .NET framework 2.0. V případě, že na našem operačním systému Windows není k dispozici, stáhneme instalátor na webové adrese:

<http://www.microsoft.com/downloads/>

Budeme-li přistupovat k vývojové databázi Oracle systému STAG, potřebujeme ještě klientský program - Oracle XE Client. Instalátor nalezneme na webových stránkách:

<http://www.oracle.com/technology/software/products/database/xe/>

Oba zmíněné instalátory jsou opatřeny průvodci s dostatečným popisem jednotlivých kroků instalace. Program po spuštění vytvoří datový soubor – data.xml. Nedoporučuje se uživateli manuálně měnit obsah tohoto souboru, mohlo by to způsobit nekonzistenci dat a jejich ztrátu.

A.7 Nastavení referenční čtečky

Níže uvedené nastavení platí pro referenční čtečku EZ One Shot model MT9060/4, ale dá se použít i pro jiné typy čteček, které emulují klávesnici. Konkrétní strany v manuálu se však budou lišit. Jednotlivé kroky nastavování se sestávají z nalezení příslušného čárového kódu a načtení tohoto kódu pomocí čtečky.

- Na titulní straně manuálu najdeme čárový kód s označením Default a načteme ho
- Na straně 19 manuálu pod oddílem „Národní rozložení klávesnice“ najdeme čárový kód s popisem „Universální“ a načteme ho
- Na straně 36 manuálu pod oddílem „Standard CODE39 FULL ASCII 39“ najdeme čárový kód s popisem „Full ASCII CODE 39 zakázat“ a načteme ho

Po nastavení je čtečka schopna plně spolupracovat s aplikací.

A.8 Použití aplikace

Prerekvizity: splnění všech kroků popsaných v kapitole o instalaci programu, čtečka čárových kódů, laserová tiskárna, archy s řezanými štítky ve formátu A4

Aplikace zatím pracuje pouze s vývojovou databází, v případě jejího „reálného nasazení“ by byl scénář použití následující:

Po spuštění aplikace se vyučující přihlásí k databázi univerzitní studijní agendy. Zvolí předmět, který vyučuje a importuje ho do lokálního uložiště. Následně vybere zkoušku ze seznamu zkouškových termínů vypsanych pro předmět, vytvoří šablonu testu a přiřadí ji k vybrané zkoušce. Pak zvolí funkci pro tisk sestavy čárových kódů. Nastaví parametry podle formátu řezaných štítků a vytiskne sestavu čárových kódů pro danou zkoušku na laserové tiskárně. Tyto akce provede až těsně před samotným zahájením zkoušky, pokud se nebude jednat o první zkoušku nebude potřeba importovat předmět. Na zkoušce pak rozdělí samolepící štítky s čárovými kódy mezi studenty, a to na základě prokázání identity studenta. Studenti si nalepí tyto štítky na přední stranu testu. Po skončení zkoušky a vyhodnocení výsledků zadá vyučující výsledky prostřednictvím čtečky čárových kódů a exportuje hodnocení pro danou zkoušku do formátu HTML. Exportovaný soubor umístí na svůj web, případně na univerzitní portál, může jej rovněž použít k vytvoření statistik v tabulkovém procesoru. Pokud bude potřeba zadat známky do studijní agendy STAG, vygeneruje sestavu se seznamem studentů a s jejich hodnocením. Tu může předat na sekretariát katedry nebo zadat známky do STAGu sám v případě, že je k tomu oprávněn.

