

Robot Karel

Robot Karel se pohybuje po obdélníkovém, šachovnicovitě uspořádaném dvorku, na jehož políčka pokládá či z nich zvedá značky. Na dvorku mohou být překážky (zdi). **Tvým úkolem je naprogramovat robota Karla, který má následující funkce** (nejlépe objektově, tzn. implementovat metody třídy Karel):

boolean dopredu(int pocetKroku) – Karel se posune o **pocetKroku** vpřed. Jít mimo dvorek, stejně jako procházet překážkami (zdi), se nedá, v případě, že bychom do toho Karla nutili, vrací funkce hodnotu **false**, pokud je možné zadaný počet kroků provést, funkce vrací hodnotu **true**.

void otocit(Karel.VLEVO) – Karel se otočí o 90° proti směru hodinových ručiček. Konstanta **Karel.VLEVO** určuje směr otáčení. v případě varianty **otocit(Karel.VPRAVO)** se Karel se otočí o 90° po směru hodinových ručiček.

boolean poloz(void) – Karel položí na políčko, na němž stojí, značku. Je-li na políčku již značka, počet značek na políčku se zvýší o jednu. Na jednom políčku může být nejvýše pět značek. Pokud se Karel pokusí položit značku na políčko, kde je již 5 značek, funkce vrací hodnotu **false**, jinak vrací hodnotu **true**.

boolean seber(void) – Karel sebere z políčka, na němž stojí jednu značku (počet značek na políčku se sníží o jednu). Není-li na políčku žádná značka, funkce vrací hodnotu **false**, jinak vrací hodnotu **true**.

Tebou vytvořená aplikace by měla umožňovat ovládání jednotlivých funkcí – každé funkci může být, například, přiřazeno tlačítko. V případě, že některá z funkcí vrátí hodnotu false, měla by na to aplikace upozornit (například nějakým zvukem, nebo vizuálně – zablikáním Karla).

Výchozí tvar dvorku, umístění Karla, překážek a značek se načte z textového souboru, který má následující formát:




D	x	y		<i>rozměry dvorku (vodorovně, svisle)</i>
K	x	y		<i>souřadnice [x, y] pozice Karla (horní levý roh má souřadnice [0, 0])</i>
P	x	y		<i>souřadnice [x, y] pozice překážky</i>
P	x	y		<i>souřadnice [x, y] pozice další překážky</i>
...				
Z	x	y	n	<i>souřadnice [x, y] políčka se značkami a jejich počet n</i>
Z	x	y	n	<i>souřadnice [x, y] dalšího políčka se značkami a jejich počet n</i>
...				

Tebou vytvořená aplikace by měla umožňovat zvolit (například pomocí komponenty OpenFileDialog) textový soubor s údaji o dvorku. Rozhodčí budou mít připraveno několik textových souborů k otestování Tvé aplikace.

Například pro soubor **dvorek.txt**, který obsahuje následující řádky

```
D 11 8
K 2 1
P 5 4
P 6 4
P 7 4
P 8 4
P 9 4
P 3 6
P 4 6
P 5 6
P 6 6
Z 8 1 3
Z 4 8 1
```

vypadá dvorek následovně (design dvorku záleží čistě na Tvé fantazii):

										
				P	P	P	P	P		
		P	P	P	P					
										

Kritéria hodnocení:

- implementace funkcí **dopředu**, **otocit**, **poloz**, **seber** (celkem **5 bodů** – 1 bod za každou fungující funkci, 1 bod za signalizaci zvukovou nebo vizuální signalizaci neprovedení některé funkce)
- správně fungující načítání dvorku ze souboru s daným formátem (**4 body**)
- kvalita programového kódu – vhodné použití technik OOP (**2 body**)
 - přehlednost kódu (**1 bod**)
- ošetření proti chybám – rozpoznání nesprávného formátu vstupního souboru (**2 body**)
- grafický vzhled aplikace (**1 bod**)

Maximální možný počet získaných bodů je 13.